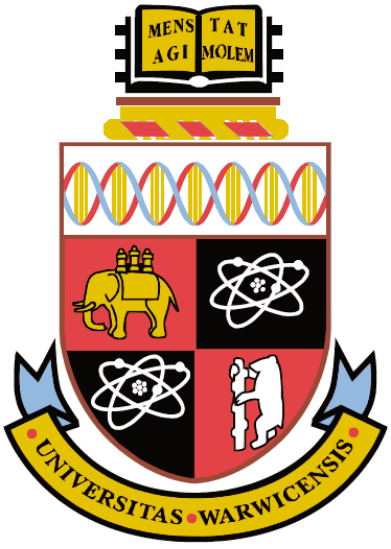
Dissertation

An algebraic characterisation of staged trees: their geometry and causal implications

Christiane Görgen

Submitted in partial fulfilment of the requirements
for the degree of **Doctor of Philosophy**

12th April 2017



Department of Statistics
University of Warwick

Contents

List of Figures

Abstract

This dissertation develops the mathematical formalism to analyse now established *staged tree models* which are graphical, discrete and parametric statistical models. We investigate the properties of these models in three important settings: geometrically, specifying these as algebraic varieties inside a probability simplex; statistically, characterising the class of all graphical representations of the same family of probability distributions; and philosophically, formulating putative causal hypotheses inferred from a class of staged trees.

Acknowledgements

First and foremost, I am immensely grateful to my supervisor Jim Smith for the excellent guidance and support. Jim has been an exceptional teacher, supervisor and collaborator who has always been available and has never ceased to encourage my efforts. His personal enthusiasm and his scientific intuition have provided me with the best start to an academic career I could have wished for.

Further major thanks go to four people who have played a central role in my young academic life: to my Italian collaborators, Manuelle Leonelli, Eva Riccomagno and Anna Bigatti, for being amazing people to work with and to Elke Thönnes for being a role model.

Second, I wish to thank the panel of my interim reports, Jon Warren and Wilfrid Kendall, for their very thorough examination of my work. The suggestions I have received from both professors have been hugely helpful in improving the presentation of my results.

Third and last, my heartfelt thanks go to the members of the Earlsdon Wheelers and of the Coventry Cyclists' Touring Club and to the great people at the Coventry Peace House for keeping me mentally sane over the past three and a half years.

Declarations

I declare that I have developed and written the enclosed thesis completely by myself and have not used sources or means without declaration in the text.

I am first author of two papers. The first paper is entitled *Equivalence Classes of Staged Trees*, throughout cited as Görgen and Smith (2015), and has been developed in close collaboration with my supervisor Jim Smith. This paper forms the core of my project and has been accepted subject to minor revisions by Bernoulli. An extended version of the results of this paper—containing a number of new examples and additional insights—is the content of Section 3.2 (Polynomial and statistical equivalence) of this thesis. The second paper, entitled *A differential approach to causality in staged trees* is here cited as Görgen and Smith (2016) and has been developed in an equally close collaboration. This work was accepted to the proceedings of the Eighth International Conference on Probabilistic Graphical Models after successfully completing a double-blind peer-review. The material of that paper forms Section 4.1 (Causal interventions on staged trees) of this thesis. In both of these publications, I have lead the preparation of the material and the conceptual work.

I am joint first author of *A Differential Approach for Staged Trees*, cited as Görgen et al. (2015). This paper has been published in the conference proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty after peer-revision. All material herein has been developed jointly with Manuele Leonelli, with Jim Smith taking a supervisory role, and has my co-author's approval for presentation in this thesis. These results can be found in Section 3.1 (A differential approach). In addition, the preprint entitled *Sensitivity analysis, multilinearity and beyond* and cited as Leonelli et al. (2015) contains joint work with Manuele Leonelli and Jim Smith that I am not the lead author of. In this thesis, I will refer to that paper exclusively for pointing out related streams of research, without repeating any of the original results.

The report entitled *Discovering statistical equivalence classes of discrete statistical models using computer algebra* is here cited as Görgen et al. (2017) and is the result of joint work with Jim Smith and our collaborators Eva Riccomagno and Anna Bigatti from the Università degli Studi di Genova, Italy. At the time of writing this report has not been submitted for publication. I

will sketch the ideas of this paper in a different form in Section 3.3 (Eliciting a graph from a tree-compatible polynomial) and I have my collaborators' approval for doing so.

Finally, I am one of three authors of the book *Chain Event Graphs*, here Smith et al. (2017), which is currently in preparation for Chapman and Hall. The three chapters I am mainly responsible for contain both new material and parts of the material cited above as well as supplementary illustrations which are not additionally published in this thesis. Chapters in that book which have been written by my co-authors are not repeated in this thesis, and relevant results which are cited here are always marked by their original source.

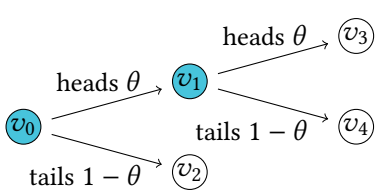
Chapter 1 (Fundamentals) and Chapter 2 (A geometric analysis of staged tree models) as well as Section 4.2 (Causal discovery in the CHDS) have at the time of writing not been published elsewhere. In particular, the material presented in Chapter 2 is entirely my own work and I am currently preparing these ideas for a collaboration with Piotr Zwiernik from the Universitat Pompeu Fabra, Spain. The two new streams of research—algebraic geometry and causal discovery—that are only touched upon in these chapters form the foundational part of work in progress which will be discussed at the very end of this thesis.

This text has been typeset using the KOMA-Script class `scrbook` of LaTeX2$_\epsilon$. All illustrations except for Figs. 2.6 and 2.7 have been created using the `tikz` and `pgfplots` packages (PGF Ti*k*Z, 2010). Figures 2.6 and 2.7 have been created using `Mathematica` (Wolfram Research Inc., 2016). All computations in commutative algebra have been performed using `CoCoA-5` (Abbott et al., 2016).

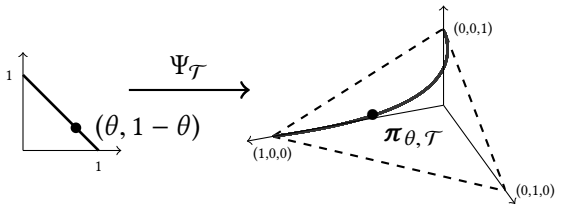
Introduction

Probability trees—or tree diagrams—are perhaps best known as useful illustrations for solving problems in high-school mathematics. For many students, these trees may be the first type of 'graphical model' to come across, and even in undergraduate probability classes a quickly drawn tree graph can help to solve tricky questions (Blitzstein and Hwang, 2014). Typical applications of probability trees include discrete experiments such as coin tossing: illustrated in Fig. 0.1. Here, the likelihood of an outcome of the experiment can simply be calculated by multiplying transition probabilities along the relevant path in the tree and possibly summing over all paths belonging to an event of interest. Despite this common application in not too advanced mathematics, the usefulness and true value of probability trees in statistical inference has, until recently, not been fully appreciated by the scientific community. Although tree graphs are frequently used in decision analysis, probability trees themselves have had considerably less interest and the standard theory of graphical models is very much dominated by Bayesian networks. In fact, it was Smith and Anderson (2008) who first successfully established a statistical framework for using probability trees as more than pretty illustrations of simple problems: the authors developed statistical methodologies based on the properties of probability trees which could be applied to inference for instance in health and social sciences. Their main contribution was to establish an elegant and self-contained toolbox to use probability trees not for embellishing other statistical models but as graphical statistical models in their own right. Now, eight years on, researchers have understood this framework so well and have used it so successfully that we are ready to go one step further. We can now use only the abstract characterisation of a model represented by a probability tree and develop statistical methods based on this, without explicitly referring back to a tree graph. In doing so, we are able to find surprising analytical results in a very general framework.

This thesis has four main objectives. First, to create a sound mathematical formalism which enables us to discuss probability tree models from a viewpoint of mathematical statistics, embedding them in established theory of parametric and discrete statistical models. Second, to interpret this formalism from an algebraic perspective, analysing a geometric characterisation of these models whose properties can be linked back to results useful in inference. Third and foremost, to characterise all probability tree representations of the same statistical model in



(0.1.1) A probability tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ which represents a repeated coin-toss model.



(0.1.2) The parametrisation $\Psi_{\mathcal{T}}$ belonging to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ from Fig. 0.1.1 maps edge labels to a point in the model (0.1).

Figure 0.1. A staged tree model on three atoms. Graphical representation in terms of a coloured probability tree and depiction as a parametric curve inside a probability simplex.

order to improve current model selection techniques. And fourth, to develop a framework in which probability trees can be given a putative causal interpretation.

The very simple example below illustrates some of the key ideas of this thesis.

Consider a coin toss which will be repeated once if the first outcome is heads. The probability of heads is assumed to be strictly positive but unknown. We can draw a probability tree denoted $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and depicted in Fig. 0.1.1 to represent this problem. The pair $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is then given by a tree graph $\mathcal{T}$, illustrating how the experiment unfolds, and a vector of labels $\boldsymbol{\theta}_{\mathcal{T}}$ whose components are conditional probabilities of events. In our example, the graph $\mathcal{T} = (V, E)$ has a vertex set $V = \{v_0, v_1, v_2, v_3, v_4\}$ and an edge set $E = \{e_1 = (v_0, v_1),\ e_2 = (v_0, v_2),\ e_3 = (v_1, v_3),\ e_4 = (v_1, v_4)\} \subseteq V \times V$. We label the edge e_1 by the probability of heads, denoted $\theta = \theta(e_1)$, and, because the transition probabilities from the same vertex shall sum to unity, the edge e_2 by the probability of tails, so $\theta(e_2) = 1 - \theta$. Because two tosses of the same coin are independent, the transition probabilities from v_0 to v_1 and v_2 are the same as from v_1 to v_3 and v_4, or rather $\theta(e_3) = \theta(e_1)$ and $\theta(e_4) = \theta(e_2)$. We always code these equations graphically by assigning the same colour to those vertices in the graph which have the same emanating edge labels: here, using blue colour in Fig. 0.1.1. Two equally coloured vertices are said to be in the same *stage*. The vector of labels $\boldsymbol{\theta}_{\mathcal{T}} = (\theta(e_1), \theta(e_2))$ of this probability tree lies inside the parameter space depicted on the left hand side of Fig. 0.1.2. Note that we will often over-parametrise a model in the way we do here. This approach will be particularly useful when subsequently ignoring sum-to-1 conditions. Now, the set of root-to-leaf paths $\Lambda(\mathcal{T}) = \{\lambda_1, \lambda_2, \lambda_3\}$ of the probability tree corresponds to the set of possible single outcomes of our experiment as follows. We identify the sequence of edges $\lambda_1 = (e_1, e_3)$ with the outcome 'heads, heads', $\lambda_2 = (e_1, e_4)$ with 'heads, tails' and $\lambda_3 = (e_2)$ with 'tails'. When multiplying edge labels along these paths, we obtain the probabilities of the respective outcomes as $\pi_{\theta,\mathcal{T}}(\lambda_1) = \theta^2$, $\pi_{\theta,\mathcal{T}}(\lambda_2) = \theta(1-\theta)$ and $\pi_{\theta,\mathcal{T}}(\lambda_3) = 1-\theta$. These can be simply read off the given graph. We

can thus specify our coin-toss model as the set of distributions $\pi_{\theta,\mathcal{T}}$ over three atoms which fulfil the criteria above, for all possible probabilities θ of 'heads':

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \left\{ \left(\theta^2, \theta(1-\theta), 1-\theta \right) \mid \theta \in (0,1) \right\}. \tag{0.1}$$

Note that all components of a vector $\boldsymbol{\pi}_{\theta,\mathcal{T}} \in \mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ in the model are strictly positive and sum to unity, and that every vector thus corresponds to a positive distribution over the three atoms. The set (0.1) of all these distributions is called a *probability tree model*. The model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ can be defined as the image of a bijective map

$$\Psi_{\mathcal{T}} : \left\{ (\theta_1, \theta_2) \in (0,1)^2 \mid \theta_1 + \theta_2 = 1 \right\} \to \mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}, \quad (\theta, 1-\theta) \mapsto \boldsymbol{\pi}_{\theta,\mathcal{T}} \tag{0.2}$$

which identifies a choice of parameters with a vector of atomic probabilities. Every probability tree thus specifies a parametrisation rule whose image is the model represented by that tree. The probability tree itself is a graphical representation of that model. Interestingly, the map (0.2) also determines a parametric curve in three-dimensional space. We draw this curve, and thus the model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ from (0.1), inside a two-dimensional (probability) simplex on the right hand side of Fig. 0.1.2.

So instead of limiting ourselves to using the probability tree merely as an elegant picture of a discrete experiment, we have already found three different viewpoints from which to analyse this example setting. First, we can characterise properties of the pair $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ as a labelled graph in the framework of graph theory and computer science. Second, we can specify a probability tree model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ as the set of probability distributions represented by such a graph $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. Third, we can define this model as the image of a certain bijective map $\Psi_{\mathcal{T}}$. The role played by the labels of a probability tree then changes according to the framework we choose: every component of a vector $\boldsymbol{\theta}_{\mathcal{T}}$ is simply a label in a symbolic framework—or an *indeterminate* in an algebraic framework—but a *parameter* in a geometric and a *probability* in a statistical framework. We unravel this subtlety in the first chapter below. Within this text, we will then combine all of these different viewpoints, and thus three usually thought of as different disciplines of mathematics, to answer some simple questions in probability tree models. In addition to the individual results achieved and presented below, in this thesis we thus also make a major contribution in linking the relatively recent study of coloured probability trees to well-known concepts across other fields of research.

Consider now Fig. 0.2 for a generic illustration of the different aspects of a tree model that we will analyse here. The tetrahedron depicted in the centre of that figure represents a *probability simplex*, the space of which all tree models are subsets. In fact, the hypersurfaces coloured in

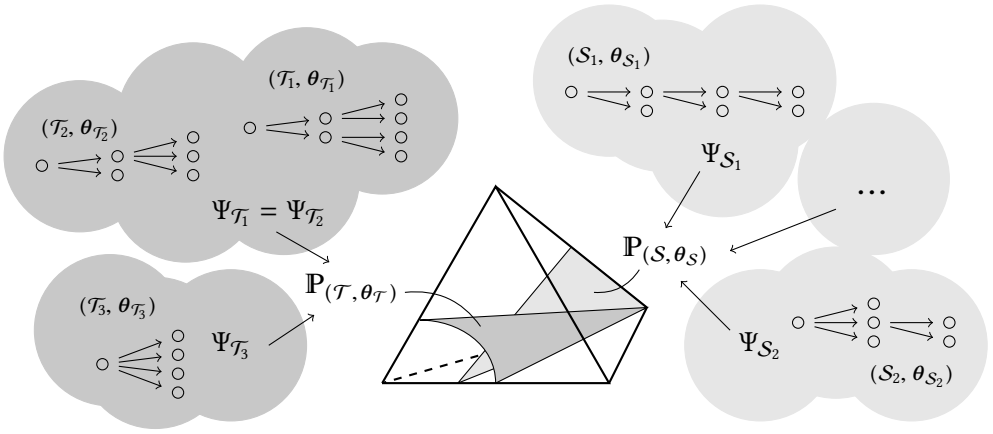


Figure 0.2. An illustration of the different aspects of probability tree models analysed in this thesis.

light and dark grey inside this space depict two different generic tree models: as sets of points, or parametric curves, just like in the example above. These models can be specified as the images of feasible parametrisations. Each such parametrisation is in turn induced by a collection of probability tree representations, depicted inside their respective cloudy shapes. The four chapters of this thesis will each analyse a different component of this picture: Chapter 1 formally defines all objects within the figure, Chapter 2 analyses algebro-geometric properties of the hypersurfaces inside the simplex, Chapter 3 characterises all probability trees which share a common parametrisation map and all parametrisations which induce the same model, and finally Chapter 4 abstracts from this picture to give a probability tree model an interpretation in an inferential context of interest. In particular, the content of these chapters is as outlined below.

In Chapter 1 of this text we introduce the probability tree model as a certain type of discrete statistical model and embed it in well-established theory of statistical methodology and Algebraic Statistics. We show how this formalism significantly tightens the initial work on probability trees. *Staged tree* and *Chain Event Graph (CEG)* models were first designed to describe discrete processes which do not follow the implicit symmetries in Bayesian network (or often for short *BN*) models. It was only later appreciated how important this class was in its own right. So Smith and Anderson (2008) introduced these alternative graphical models which turned out to be both easy to communicate to a lay audience and could fit observational data better than more standard models. The staged tree is a coloured probability tree whose colouring enables the expert to read off numerous context-specific conditional independence statements between depicted events. In a sense that we formalise below a CEG then provides a

more compact graph based on a staged tree which allows for the design of efficient propagation algorithms and model selection techniques. We repeat the respective definitions and give an overview of the relevant results in the main body of the text.

Whilst early publications in this field used the terminology 'staged tree' and 'staged tree model' exchangeably, we now appreciate the necessity to distinguish between a graphical *representation* of a statistical model and the model itself, which is a collection of probability distributions with certain properties. This distinction is common practice in the theory of Bayesian networks where *acyclic digraphs* or *Markov random fields* are graph representations of an underlying statistical model (Lauritzen, 1996). These representations code conditional independence assumptions between a set of random variables in a network where problem variables are vertices and the absence of edges between vertices indicates (conditional) independence assumptions. The set of all distributions which respect these assumptions then determines the corresponding statistical model, a Bayesian network model. We show that every discrete BN model can be represented by a staged tree or CEG but that in many problems CEGs are much more expressive.

In Chapter 2, we make first advances in an analysis of the geometric properties of staged tree models. Here, we always specify such a model as the image of a parametrisation map. For instance in the toy example above, this image was simply given by a parametric curve. In much more generality, staged tree models can be equivalently characterised as solution sets of systems of very specific polynomial equations. Ours models are hence algebraic *varieties* or rather *semi-algebraic sets* when imposing positivity constraints on the immanent probability distributions. The main purpose of this chapter is to infer the nature of the equations defining these sets and to present a number of small-scale examples. Intriguingly, we find in Theorem 1 that the system of equations characterising the stage structures in a coloured probability tree always corresponds to relationships between *odds ratios* of probabilities of events. These equations allow thus for a very straightforward interpretation in terms of the underlying statistical model. Just like a coloured identification of edge labels in a staged tree, these odds-ratio equations can also be simply read from a tree graph.

We then present a complete analysis of all staged tree models on three or four atoms, drawing these as subsets of probability simplices: compare Figs. 0.1 and 0.2 above. Throughout this chapter, we list methods from algebraic geometry which can be employed in this type of study of a statistical model and outline the challenges which arise when analysing a probabilistic object within an algebraic framework.

Chapter 3 forms the core of this thesis. Here, our main aim is to deduce a result for staged tree models which is analogous to the concept of Markov equivalence for Bayesian networks. To

this end we define two staged trees to be *statistically equivalent* if and only if they represent the same probability tree model—so if they are in a cloud of the same colour as the corresponding surface in Fig. 0.2. We will then solve two problems which are central to statistical inference:

1. How can we classify all staged trees representing the same model?
2. Are every two members within a class of statistically equivalent model representations connected by simple 'local' operations?

Thwaites and Smith (2015b) have shown that the question of statistical equivalence in staged tree models can unfortunately not be answered in a purely graphical fashion, as is the case in BN models. However, using the machinery developed in Chapter 1 and Görgen and Smith (2015), we will show that every probability tree is in one-to-one correspondence with a certain factorisation of a polynomial in the edge labels of that graph. We call this polynomial the *interpolating polynomial* and derive that all graphs belonging to the same model then share that same interpolating polynomial, up to possible reparametrisations. Those that share a common polynomial description are in the same cloudy shape in Fig. 0.2 while those that need to be reparametrised are in different clouds pointing towards the same hypersurface in that figure. This answers question (1.) above. Interestingly, the interpolating polynomial can then also be used to answer question (2.). In fact, an application of the distributive law on a certain 'nested' form of an interpolating polynomial corresponds to a local change of a subgraph in the staged tree. A series of reorganisations of the nesting together with substitutions of factors in the polynomial is then shown to be analogous to what we call *swap* and *resize* operations in the graph. That these can traverse the full equivalence class is the key result of Theorem 2.

We conclude this development with an application of the swap and resize operators on the statistical equivalence class of a staged tree fitted to a real dataset.

In the same chapter, we find a number of other useful features of the interpolating polynomial. These are its use in calculating marginal and conditional probabilities in staged tree models using a differential operator as developed in Görgen et al. (2015) and its role in an algorithmic elicitation of all members in a given equivalence class of staged trees using computer algebra as in Görgen et al. (2017).

In Chapter 4, we provide a causal interpretation for the findings of Chapter 3. This objective has long been desired by the community of researcher in chain event graph models and has already been approached in various less successful (because graph-based) attempts. We are now ready to develop an unambiguous language for causal inference in these models, using the newly developed algebraic characterisation. We first show how the interpolating polynomial can be used in a differential framework to express causal manipulation operations as presented in Görgen and Smith (2016). Then, we note that specifying all the different representations of

the same model using an interpolating polynomial allows us to specify all different orders in which events can be depicted across a statistical equivalence class. So following Pearl (2000) we assert that if there are two graphical model representations, one stating that an event A can happen before a different event B and the other representation reversing that order to B happening before A, then we would not want to consider A as a possible cause of B or vice versa. However, if there is an unambiguous ordering across all possible staged tree representations, then we might call one event a *putative cause* of the other and employ causal inference techniques to measure the strength of that causal effect. We show how this notion can be formalised for staged trees.

Finally, we employ the developed techniques to analyse the causal hypotheses drawn out of the staged tree model inferred from a real dataset that we will have introduced in the preceding chapter.

1. Fundamentals

In this first chapter we will start off by introducing very general probability trees. These trees can provide elegant graphical representations of certain parametric statistical models. In particular, via a colouring of their vertices, they can code a number of conditional independence assumptions: a tool frequently used in describing real systems. We illustrate briefly that such coloured probability trees—called *staged* trees—can be specified either by imposing linear constraints on their parameter space or by finding polynomial constraints on certain probabilities of events in an underlying space. Deferring a deeper analysis of this observation to the subsequent chapter, we then show that every discrete Bayesian network model can be represented by a staged tree. A small-scale example simplifying a real system illustrates this point. We explain why a graphical representation in terms of a probability tree can often be much more expressive than the more standard choice of an acyclic digraph. We then end the chapter introducing some vocabulary from algebraic geometry and polynomial algebra which can be naturally used to capture properties of staged trees. A translation of notions from algebra to staged trees and back will be central to the development in this thesis.

Throughout, we assume a basic knowledge of Bayesian networks (BNs) in order to be able to compare our new tools with those which are well known in graphical models. The terminology *Bayesian network model*—or for short BN model—hereby refers to a set of probability distributions with certain features, and the term *acyclic digraph* refers to a graphical representation of that set, given by a directed graph with no directed cycles. All further concepts needed will be very briefly repeated below. A thorough introduction to BN models can be found in Lauritzen (1996) and a more applied approach is presented for instance in Smith (2010).

1.1. Parametric statistical models

A statistical model, as defined below, is simply a set of probability distributions over a given space (Blitzstein and Hwang, 2014). If that space is countable, we say that such a model is *discrete*. Every distribution in a finite and discrete statistical model can be specified as the vector of the values it takes over the elements of the underlying space. Then each component of this vector is a value between zero and one, and the sum of all components is equal to one.

Following Drton et al. (2009), discrete statistical models are thus sets of vectors with the above property, so sets of points lying inside a *probability simplex*

$$\Delta_{n-1} = \left\{ p \in \mathbb{R}^n \mid \sum_{i=1}^{n} p_i = 1, \text{ and } 0 \leq p_i \leq 1 \text{ for all } i = 1, \ldots, n \right\} \tag{1.1}$$

where $n \in \mathbb{N}$ is the cardinality of the discrete space. When distributions are assumed to be strictly positive, they are elements of the *open* probability simplex $\Delta^{\circ}_{n-1} \subseteq \Delta_{n-1}$ where $p_i \in (0, 1)$ for all $i = 1, \ldots, n$. For the purpose of this thesis we often assume openness of probability simplices. This assumption enables us to avoid distracting technical issues concerning boundary cases and so to focus more directly on inferential matters.

Statistical models can often be characterised using a set of parameters and may then be defined as families of distributions together with certain constraints on these parameters, for instance constraints on the mean and the variance in Gaussian models. Whenever this is the case it is vital to be able to uniquely identify each parameter in a space with one distribution in the model. We will thus use the following definition:

Definition 1.1 (Parametric statistical model)**.** Let Ω always denote a finite space with $n \geq 2$ *atoms* $\omega \in \Omega$, and let $\Theta \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$, denote a parameter space. We write $p_{\boldsymbol{\theta}} : \Omega \to (0, 1)$ for a strictly positive probability mass function over the atoms of such a finite space, and always assume that $p_{\boldsymbol{\theta}}$ can be parametrised using some $\boldsymbol{\theta} \in \Theta$. The vector of values of that function is denoted by the bold character $\boldsymbol{p}_{\boldsymbol{\theta}} = \big(p_{\boldsymbol{\theta}}(\omega) \mid \omega \in \Omega\big)$. Henceforth we call each of the components $p_{\boldsymbol{\theta}}(\omega)$, $\omega \in \Omega$, of that vector an *atomic probability*.

A *discrete parametric statistical model* on Ω is a set of vectors as above

$$\mathbb{P}_{\Psi} = \left\{ \boldsymbol{p}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta \right\} \subseteq \Delta^{\circ}_{n-1} \tag{1.2}$$

which lie in the open $n-1$-dimensional probability simplex (1.1) for $n = \#\Omega$. The index Ψ in $\mathbb{P}_{\Psi}$ is a bijective map

$$\Psi : \Theta \to \mathbb{P}_{\Psi}, \quad \boldsymbol{\theta} \mapsto \boldsymbol{p}_{\boldsymbol{\theta}} \tag{1.3}$$

which uniquely identifies a choice of parameters with a distribution in the model. Ψ is called a *parametrisation* of the model $\mathbb{P}_{\Psi} = \Psi(\Theta)$.

An illustration of the definition above is given in Fig. 1.1. We will use this slightly non-standard definition of a parametric model because the development of later chapters requires an accurate differentiation between a model as a parametrised set of points and the parametrisation as a map with certain properties. In particular, in Chapter 2 we present a characterisation first of the domain of a parametrisation of a statistical model and then of its image, so

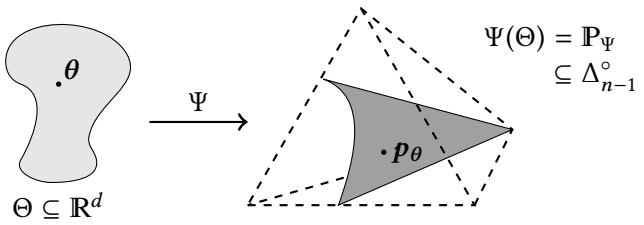


Figure 1.1. A discrete and parametric statistical model $\mathbb{P}_\Psi = \Psi(\Theta)$ depicted in dark grey as a subset of the $n-1$-dimensional open probability simplex Δ°_{n-1}, here illustrated as the hatched tetrahedron. The model equals the image of a parametrisation $\Psi : \Theta \to \Delta^\circ_{n-1}$, $\boldsymbol{\theta} \mapsto \boldsymbol{p}_{\boldsymbol{\theta}}$ with parameter space $\Theta \subseteq \mathbb{R}^d$ depicted in light grey.

the model itself. Also note in this context that we use the blackboard bold letter $\mathbb{P}$ for sets of probability distributions p, rather than for a probability measure—here instead denoted P—as is often the case in the literature. Bold symbols such as $\boldsymbol{p}$ or $\boldsymbol{\theta}$ always indicate vectors, and calligraphic symbols will be reserved for graphs.

We will henceforth restrict our analysis to discrete models which can always be parametrised. So we always refer to a discrete and parametric statistical model when using the term 'model'.

In the symbolic and algebraic framework used in later sections, instead of specifying a parameter space $\Theta \subseteq \mathbb{R}^d$ we will often interpret a parametric model as a set of points whose components can be expressed using *indeterminates* $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$. We would then choose one symbolic vector $\boldsymbol{p}_{\boldsymbol{\theta}}$ to represent a set $\{\boldsymbol{p}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$. For instance, in the example from page 2 this would amount to representing the coin-toss model by a vector $\boldsymbol{p}_{\boldsymbol{\theta}} = (\theta_1^2, \theta_1\theta_2, \theta_2)$ whose components are products of indeterminates $\boldsymbol{\theta} = (\theta_1, \theta_2)$ instead of substituting values such that these indeterminates are probabilities, $(\theta_1, \theta_2) \in \Delta^\circ_{2-1} \subseteq \mathbb{R}^2$.

The discipline of Algebraic Statistics has successfully followed such a symbolic approach to inference and has used it to analyse the algebraic and geometric properties of a number of interesting statistical models, mostly discrete or Gaussian: see the detailed discussion in Chapter 2. In this development, a statistical model is often characterised using polynomial equalities (and inequalities) of one of two types. The first type constrains atomic probabilities, so here we would have a collection of $f_i(\boldsymbol{p}) = 0$ where a probability distribution $\boldsymbol{p} \in \Delta_{n-1}$ functions as a vector of indeterminates and f_i is a polynomial for all $i = 1, \ldots, k$. Alternatively, models are often specified in terms of constraints on their parametrisation, so $g_i(\boldsymbol{\theta}) = 0$ for polynomials g_i and indeterminates which are parameters $\boldsymbol{\theta} \in \Theta$, $i = 1, \ldots, k$. So in Fig. 1.1, we would specify a model by imposing constraints either the set on the right hand side using the first option or, equivalently, the set on the left hand side for the second. In order for these

constraints to be of polynomial form, model parametrisations are often assumed to be rational functions. In the type of models we introduce below, sometimes polynomial constraints on atomic probabilities can be equivalently expressed as polynomial constraints on parameters and vice versa:

Definition 1.2 (Monomial parametrisation)**.** Let $\mathbb{P}_\Psi$ be a parametric model as in Definition 1.1. We say that the map $\Psi : \Theta \to \mathbb{P}_\Psi$ is a *monomial parametrisation* if every component of its image is a monomial

$$\Psi_i(\boldsymbol{\theta}) = \theta_1^{\alpha_{i,1}} \cdots \theta_d^{\alpha_{i,d}} \tag{1.4}$$

in the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d) \in \Theta$ with non-negative exponent $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,d}) \in \mathbb{Z}^d_{\geq 0}$ for $i = 1, \ldots, n$ and $d \in \mathbb{N}$. A monomial parametrisation is called *multilinear* if the multiplicity of every parameter in every such monomial is at most one, $\alpha_i \in \{0, 1\}^d$ for all $i = 1, \ldots, n$. We then also call the model $\mathbb{P}_\Psi$ a *multilinear model.*

A monomial expression like (1.4) is often called a *power product* (Pistone et al., 2001a) in Algebraic Statistics: see Section 1.3. Perhaps more familiar to the reader than the power-product representation of a monomial model is its alternative reparametrised form as an exponential family

$$\Psi_i(\boldsymbol{\phi}) = \exp\left(\sum_{j=1}^{d} \alpha_{i,j}\phi_j\right) \tag{1.5}$$

whose parameters $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_d)$ are given by $\phi_j = \log(\theta_j)$ and $\theta_j \in (0, 1)$, $j = 1, \ldots, d$. Incidentally, because of this equivalence the tree models we define in the next section can be viewed as exponential families or log-linear models. In this thesis we will usually prefer the expression (1.4)—which further enables us to interpret multilinear models as certain families of multinomial distributions—over (1.5). This is in order to avoid having to deal with positivity conditions of parameters and in order to be able to develop a theory which links the edge labels of a graph directly to a polynomial representation. This theory in turn has then some surprisingly natural links to other areas of mathematics as we will point out in the introduction to Chapter 3.

Monomial parametrisations commonly occur in statistical modelling. For instance, in every Bayesian network model atomic probabilities are of the type $p_{\boldsymbol{\theta}}(\omega_i) = \theta_1^{\alpha_{i,1}} \cdots \theta_d^{\alpha_{i,d}}$ where $i = 1, \ldots, n$. In this case, the θ-parameters are conditional or marginal probabilities: see also the development below. If the model contains no loops or repetitions—it is not *dynamic*—these parametrisations are also always multilinear. The importance of general classes of multilinear models has only recently been fully appreciated: see Leonelli et al. (2015) and references given therein.

Consider now an example of a multilinear model which can be completely characterised by a collection of very special polynomial constraints.

Example 1.3 (Cross-product differences)**.** We briefly recall a setting analysed in Geiger et al. (2006). Let Ω be a discrete space and let $X = (X_1, X_2, X_3) : \Omega \to \{0,1\}^3$ be a vector of random variables measurable with respect to that space, with a strictly positive probability distribution $p : \{0,1\}^3 \to (0,1)$. We denote the values taken by that distribution by the shorthand $p_{x_1x_2x_3} = p(x_1, x_2, x_3)$ using subscripts for function arguments $x_i = 0, 1$ and $i = 1, 2, 3$; and we denote the corresponding vector of atomic probabilities by $\boldsymbol{p} = (p_{x_1x_2x_3} \mid x_i = 0, 1,\ i = 1, 2, 3)$. This vector is an element of the open seven-dimensional probability simplex, $\boldsymbol{p} \in \Delta^\circ_{8-1}$, because its eight entries sum to one and are positive. It is easy to check[1] that a statement of the type 'X_1 is independent of X_3 given X_2'—in symbols $X_1 \perp\!\!\!\perp X_3 \mid X_2$—translates into what the authors of the publication cited above call *cross-product differences* on the atomic probabilities. These are polynomial constraints of the form

$$f_1(\boldsymbol{p}) = p_{000}p_{101} - p_{001}p_{100} = 0 \quad \text{and} \quad f_2(\boldsymbol{p}) = p_{010}p_{111} - p_{011}p_{110} = 0. \tag{1.6}$$

So the discrete conditional-independence model we analyse here can be specified as the intersection of the set of all points which both fulfil (1.6) and also lie within a probability simplex:

$$\mathbb{P} = \left\{\boldsymbol{p} \in \mathbb{R}^8 \mid f_1(\boldsymbol{p}) = 0 \text{ and } f_2(\boldsymbol{p}) = 0\right\} \cap \Delta^\circ_{8-1}. \tag{1.7}$$

In the language of Chapter 2, the model thus equals the intersection of a toric variety with a linear affine hyperplane and a semi-algebraic set.

We will now show here that—different from the approach followed by the authors cited above—$\mathbb{P}$ can equivalently be specified as a parametric model as in Definition 1.1. This is because the conditional independence constraint above can be expressed by the acyclic digraph $X_1 \to X_2 \to X_3$ and we can write the distribution $p_{x_1x_2x_3} = \theta_{x_1x_2}\theta_{x_1x_2x_3}$ in form of a recursive factorisation according to that graph, so as the product of two parameters $\theta_{x_1x_2} = p(x_1, x_2)$ and $\theta_{x_1x_2x_3} = p(x_3|x_1, x_2)$ which are marginal and conditional probabilities, for $x_i = 0, 1$ and $i = 1, 2, 3$. Then the conditional independence assumption above translates into the polynomial constraints

$$g_{jk}(\boldsymbol{\theta}) = \theta_{0jk} - \theta_{1jk} = 0 \quad \text{for } j, k = 0, 1 \tag{1.8}$$

1 See for instance Example 1.18.

where $\boldsymbol{\theta} = (\theta_{00}, \ldots, \theta_{111})$. Hence, the model (1.7) is a parametric model $\mathbb{P} = \mathbb{P}_\Psi$ with multilinear parametrisation

$$\begin{aligned} \Psi : \quad & \left(\Delta^\circ_{4-1} \times \Delta^\circ_{8-1}\right) \cap \left\{\boldsymbol{\theta} \in \mathbb{R}^{12} \mid g_{jk}(\boldsymbol{\theta}) = 0 \text{ for } j,k = 0,1\right\} \to \Delta^\circ_{8-1} \\ & (\theta_{00}, \theta_{01}, \theta_{10}, \theta_{11}, \theta_{000}, \theta_{001}, \ldots, \theta_{111}) \mapsto (\theta_{ij}\theta_{ijk} \mid i,j,k = 0,1) \end{aligned} \tag{1.9}$$

such that the image of Ψ equals the model itself.

There are two important points to note here. First, the constraints in (1.6) are polynomial whereas the new constraints in (1.8) are linear. So our alternative representation is both algebraically and geometrically much simpler than the original cross-product differences. However, the linear constraints have the drawback that they depend on a chosen parametrisation of the model. Similarly, and second, whilst (1.9) expresses the properties of our model via a constraint on the domain of a chosen parametrisation, (1.7) expresses these equivalently as constraints on the image of that parametrisation: compare again Fig. 1.1. We will learn in Chapter 2 that in the type of models we are most interested in, linear constraints on the domain of a parametrisation always translate into polynomial constraints on the image and that both of these characterisations are equivalent.

We will come back to the example above in subsequent sections. Despite its simplicity, it enables us both to discuss an interesting parametric model in a rigorous mathematical fashion with a view on possible algebraic characterisations of the properties of the underlying probability distributions, and it can be represented by a set of probabilities trees which can be analysed using that algebraic characterisation. In fact, the for us most interesting models are those parametric models whose parametrisation can be read from a certain graph structure. These are also termed *graphical models*, or here in particular *tree models* because the graph we are interested in is (almost) always a tree graph.

1.2. Tree models

We will firstly recall some notation in order to be able to unambiguously discuss objects from graph theory. We then define models which can be represented by certain graphs, and we formally introduce parametric statistical models with that property. Centrally, our graphs can be enhanced with an additional feature—a colour—to capture key modelling assumptions.

1.2.1. Probability trees

The following definition is based on the graphical notions introduced in Shafer (1996), adapting notation from Görgen and Smith (2015).

Definition 1.4 (Event tree)**.** A finite graph denoted $\mathcal{T} = (V, E)$ with vertex set V and edge set $E \subseteq V \times V$ is called a *tree* if it is connected and has no cycles. In a *directed* tree, each edge $e = (v, v') \in E$ is a pair of ordered vertices. We call vertices $\mathrm{pa}(v) = \{v' \mid \text{there is } (v', v) \in E\}$ the *parents* of $v \in V$ and $\mathrm{ch}(v) = \{v' \in V \mid \text{there is } (v, v') \in E\}$ the set of *children* of $v \in V$. A vertex $v_0 \in V$ without parents is called a *root* of the tree and vertices without children are called *leaves*. We use the symbol λ and the term *root-to-leaf path* for a directed sequence of edges $E(\lambda) \subseteq E$ where one edge $e = (v, v')$ precedes another $e' = (w, w')$ only if $v' = w$, and where the first edge in the sequence emanates from the root and the final edge terminates in a leaf. A *subpath* in a directed tree is then a connected subsequence of a root-to-leaf path. We call a directed tree an *event tree* if all vertices except for one unique root have exactly one parent and each parent which is not a leaf has at least two children.

We denote the set of all root-to-leaf paths of an event tree by $\Lambda(\mathcal{T})$. The power set of the set of root-to-leaf paths is called the *path sigma-algebra* of the tree, denoted $\sigma(\mathcal{T})$. For fixed $v \in V$ and $e \in E$ we define *vertex-* or *edge-centred events* in the path sigma-algebra as the set of root-to-leaf paths passing through that vertex or edge, so

$$\Lambda(v) = \Big\{\lambda \in \Lambda(\mathcal{T}) \mid \text{there is an edge } (\cdot, v) \in E(\lambda)\Big\}, \tag{1.10.1}$$

$$\Lambda(e) = \Big\{\lambda \in \Lambda(\mathcal{T}) \mid e \in E(\lambda)\Big\}. \tag{1.10.2}$$

We set $\Lambda(v_0) = \Lambda(\mathcal{T})$.

A graph $\mathcal{T}' = (V', E')$ where $(V', E') \subseteq (V, E)$ is called a *subtree* of $\mathcal{T} = (V, E)$ if it is an event tree. We then write $\mathcal{T}' \subseteq \mathcal{T}$. We say that a subtree $\mathcal{T}(v) \subseteq \mathcal{T}$ is an *induced* subtree if $v \in V$ and if the root-to-leaf paths $\Lambda(\mathcal{T}(v))$ of this subtree coincide precisely with root-to-leaf paths in the event $\Lambda(v) \subseteq \Lambda(\mathcal{T})$ centred at v, except for the subpath from v_0 to v in the original tree $\mathcal{T}$. We call a subtree $(\{v\} \cup \mathrm{ch}(v), E(v)) \subseteq \mathcal{T}$ whose root-to-leaf paths are single edges $E(v) = \{(v, v') \in E \mid v' \in \mathrm{ch}(v)\}$ emanating from the same vertex $v \in V$ a *floret*, henceforth denoted by the shorthand $\mathcal{F}_v = (v, E(v))$.

An illustration of the different concepts of the above definition is provided in Fig. 1.2.

When representing a tree model as defined below, an event tree $\mathcal{T} = (V, E)$ depicts all possible unfoldings of events within a system. In particular, every root-to-leaf path $\lambda \in \Lambda(\mathcal{T})$ then corresponds to an atom in an induced sample space and depicts one possible history of a unit in the system. Every vertex $v \in V$ denotes a state that a unit following a root-to-leaf path $\lambda \in \Lambda(v)$ might find itself in, and every edge $e = (v, v')$ denotes the possibility of passing from one situation v to the next v'. The sets $\Lambda(v)$ or $\Lambda(e)$, for $v \in V$ and $e \in E$, of all paths going through one fixed vertex or edge are the sets of all atoms for which the event associated with v or e happened. These events are also called *Moivrean events* in Shafer (1996) and play a

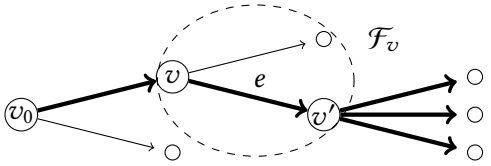


Figure 1.2. An event tree $\mathcal{T} = (V, E)$ with root $v_0 \in V$ and non-leaf vertices $v, v' \in V$ which are connected by the edge $e = (v, v') \in E$. The floret $\mathcal{F}_v$ is encircled. The thick depicted edges correspond to root-to-leaf paths in the edge-centred event $\Lambda(e)$. Note that this is equal to the vertex-centred event $\Lambda(v')$. The induced subtree $\mathcal{T}(v) \subseteq \mathcal{T}$ is given by the two florets $\mathcal{F}_v$ and $\mathcal{F}_{v'}$, and the induced subtree $\mathcal{T}(v') = \mathcal{F}_{v'}$ is a single floret. For simplicity, leaf vertices are often not named.

central role in the analysis of event trees, as we will see below. Note for instance that Moivrean events $\Lambda(v)$ are in one-to-one correspondence with induced subtrees $\mathcal{T}(v)$ if we 'condition on' arriving at their root $v \in V$. This graphical property is reflected in the calculation of the probabilities of these events: see Sections 2.1 and 3.1. Naturally, root-to-leaf paths are in one-to-one correspondence with leaf-centred events. We prefer to think about these as sequences of edges because we are often interested in the entire history of a unit rather than in its final state. Furthermore, because the graphs we examine are directed, they allow for an analysis of that directionality and an induced order on vertex- or edge-centred events. So event trees are particularly expressive in terms of an ordering of events, rather than of random variables as is the case in acyclic digraphs. As Thwaites and Smith (2015b) noted, unlike for Bayesian network models, random variables are in fact rather artificial primitives to use in tree models, and alternative event-based semantics suggest themselves. We will see examples of this below and refer to Smith et al. (2017) for further details.

We have developed the following formal definition in order to enable us to equip event trees with a probability distribution.

Definition 1.5 (Labelled event trees and probability trees)**.** Let $\mathcal{T} = (V, E)$ be an event tree and assume there are labels $\theta(e) = \theta(v, v')$ associated to all edges $e = (v, v') \in E$. We call the vector of all labels on edges emanating from a common vertex a vector of *floret labels*, denoted $\boldsymbol{\theta}_v = \big(\theta(e) \mid e \in E(v)\big)$. Then the vector $\boldsymbol{\theta}_{\mathcal{T}} = (\boldsymbol{\theta}_v \mid v \in V)$ denotes all labels associated to the event tree[2]. The pair $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ of event tree graph together with a vector of labels is henceforth called a *labelled event tree*.

We will call a labelled event tree whose labels can take real values a *probability tree* if all vectors of floret labels lie within probability simplices, so $\boldsymbol{\theta}_v \in \Delta^{\circ}_{\#E(v)-1}$ for all $v \in V$. Then

2 We usually do not a priori assume a particular order on the entries of this vector. Note also that if v is a leaf then $\boldsymbol{\theta}_v$ is empty.

the vector of all labels $\boldsymbol{\theta}_{\mathcal{T}}$ is an element of a parameter space which is a product of probability simplices, denoted $\Theta_{\mathcal{T}} = \bigtimes_{v \in V} \Delta^{\circ}_{\#E(v)-1}$. In probability trees, we call every label $\theta(e)$, $e \in E$, a *primitive probability*.

Subtrees of probability trees with inherited edge labels are called *(probability) subtrees*.

In applications in statistical modelling, we can interpret a label $\theta(v, v')$ on an edge $e = (v, v') \in E$ in a probability tree as the *transition probability* of passing from a situation v to a situation v' along that edge in the tree. A vector of floret labels $\boldsymbol{\theta}_v$ can then be thought of as a vector of conditional transition probabilities. Importantly, the constraint that floret labels shall be elements of probability simplices ensures that primitive probabilities are positive and those belonging to the same floret sum to unity: so $\sum_{e \in E(v)} \theta(e) = 1$ and $\theta(e) \in (0, 1)$ for all non-leaves $v \in V$ and $e \in E$.

Note that because of the interpretation above, edge labels $\theta(e)$ were formerly denoted as conditional probabilities $\pi_e(v'|v)$ (Smith and Anderson, 2008). However, within this thesis we often prefer to think of primitive probabilities as unnormalised *potentials* which can be marginal or conditional probabilities as in Lauritzen (1996) and whose meaning can be inferred from their positioning within a tree graph. This flexibility is key when discussing statistical equivalence in Chapter 3. Here, two probability trees representing the same model might have the same labels but their respective interpretation—and floret sum-to-1 conditions—can be very different.

Labelled event trees will be important later in this text when we interpret the labels of an event tree simply as indeterminates which are unknown, or not assigned any values. We can then determine whether such a labelled event tree can in fact be a probability tree (Chapter 3) and how the assignment of different values to these labels can change our understanding of the context the tree represents, both in a statistical and in a geometric sense (Chapter 2).

As noted above, probability trees are highly expressive in visualising how events in a discrete setting might evolve. This had first been outlined in the seminal work of Shafer (1996) where the depicted order of Moivrean events could be given a causal interpretation. However, probability trees are a lot less commonly used in statistical inference: compare the references given on page 19 below. We will now introduce the formalism elaborated in Görgen and Smith (2015) which will link Definitions 1.1 and 1.5 and will enable us to use probability trees as representations of certain parametric statistical models. This new formalism is a lot tighter than initial work on these models which did not distinguish between the statistical model itself and its graphical representation, or between different interpretations of edge labels. However, without these new tools it would not have been possible to develop many of the powerful res-

ults we present in this thesis. In particular, using the notions below our tree models can for the first time be proven to be interpretable as parametric statistical models.

Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a labelled event tree with a graph $\mathcal{T} = (V, E)$ and associated labels $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathcal{T}}$ as in the definition above. We always denote the product of all labels along a root-to-leaf path $\lambda \in \Lambda(\mathcal{T})$ by

$$\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \prod_{e \in E(\lambda)} \theta(e). \tag{1.11}$$

Henceforth, we will call the monomial in (1.11) an *atomic monomial*. We show now that if the labelled event tree is a probability tree, then atomic monomials are atomic probabilities. As a consequence, $\pi_{\boldsymbol{\theta},\mathcal{T}}$ defines a strictly positive probability mass function. This in turn induces a probability measure on a probability space associated to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. We can therefore interpret an assignment of values to the indeterminates in (1.11) as a monomial parametrisation as given in Definition 1.2. The image of that parametrisation is a statistical model which can be graphically represented by the probability tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. Explicitly, we have the following result:

Proposition 1.6 (Probability measures on a probability tree)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a probability tree and $\pi_{\boldsymbol{\theta},\mathcal{T}}$ as in (1.11). The map*

$$\begin{aligned} \Pi_{\boldsymbol{\theta},\mathcal{T}} : \quad & \sigma(\mathcal{T}) \to [0,1] \\ & A \mapsto \sum_{\lambda \in A} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \sum_{\lambda \in A} \prod_{e \in E(\lambda)} \theta(e) \end{aligned} \tag{1.12}$$

is a probability measure. The triple $(\Lambda(\mathcal{T}), \sigma(\mathcal{T}), \Pi_{\boldsymbol{\theta},\mathcal{T}})$ of set of root-to-leaf paths, path sigma-algebra and the measure above thus forms a discrete probability space, represented by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

Proof. By Definition 1.5, primitive probabilities are strictly positive so $\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) \in (0,1)$ is a positive probability for every root-to-leaf path $\lambda \in \Lambda(\mathcal{T})$. Moreover, Definition 1.5 ensures that $\sum_{e \in E(v)} \theta(e) = 1$ for all $v \in V$. Substituting these subsums into the sum of all atomic probabilities[3], we obtain that $\sum_{\lambda \in \Lambda(\mathcal{T})} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = 1$. Because $\mathcal{T}$ is a finite graph, finite additivity is sufficient to prove the claim. □

We therefore have the following definition.

Definition 1.7 (Tree model)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a probability tree, $\mathcal{T} = (V, E)$. Denote the associated vector of atomic probabilities by the bold symbol $\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} = \left(\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) \mid \lambda \in \Lambda(\mathcal{T})\right)$. Then the set

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \left\{ \boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} \mid \boldsymbol{\theta} \in \Theta_{\mathcal{T}} \right\} \subseteq \Delta^{\circ}_{\#\Lambda(\mathcal{T})-1} \tag{1.13}$$

3 This can be very easily done using a nested form of summation as in (3.23) on page 85.

is a discrete and parametric statistical model with parameter space $\Theta_{\mathcal{T}} = \bigtimes_{v \in V} \Delta^{\circ}_{\#E(v)-1}$. We call (1.13) a *(probability) tree model* and say that the elements in $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ are distributions which *factorise according to* $\mathcal{T}$. The model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ can be parametrised by the bijective map

$$\begin{aligned} \Psi_{\mathcal{T}} : \bigtimes_{v \in V} \Delta^{\circ}_{\#E(v)-1} &\to \Delta^{\circ}_{\#\Lambda(\mathcal{T})-1} \\ \left(\boldsymbol{\theta}_v \mid v \in V\right) &\mapsto \left(\prod_{e \in E(\lambda)} \theta(e) \mid \lambda \in \Lambda(\mathcal{T})\right) \end{aligned} \tag{1.14}$$

for which $\Psi_{\mathcal{T}}(\bigtimes_{v \in V} \Delta^{\circ}_{\#E(v)-1}) = \mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$. We call (1.14) a *tree parametrisation*.

The nomenclature above has been chosen in analogy to the terminology used in Lauritzen (1996) for Bayesian networks where distributions factorise according to an acyclic digraph. Note that we always index a tree model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ by one graphical representation $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ rather than by the associated tree parametrisation $\Psi_{\mathcal{T}}$. This is because the same parametrisation might arise from different graphs, so maybe $\Psi_{\mathcal{T}} = \Psi_{\mathcal{S}}$ and $\boldsymbol{\theta}_{\mathcal{T}} = \boldsymbol{\theta}_{\mathcal{S}}$[4] even though $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \neq (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. Compare Fig. 0.2 and see Section 3.2 for details on this subtlety.

Many well-known statistical models are based on underlying tree descriptions. This can be in form of special acyclic digraphs which are trees with hidden variables, often applied to problems in phylogenetics (Zwiernik, 2016), or probability decision graphs which extend the use of probability trees to settings which model decisions as well as uncertainty (Jaeger, 2004) or influence diagrams of tree form which represent a context of interest (Shachter, 1998; McAllester et al., 2008). However, for many of these authors, tree graphs have vertices labelled by random variables rather than by events: so very different from the type of tree models we defined above. Probability trees are rarely thought of as graphical models in their own right but rather as elegant representations to communicate a collection of model assumptions, often also to complement other inferential techniques: see for instance the work of Salmerón et al. (2000). None of the authors cited here introduce tree models as explicitly based on a probability tree and none of the notions developed in the references above can embed conditional independence assumptions graphically. So the framework we develop here is relatively new to the literature, with a first major publication less than a decade ago (Smith and Anderson, 2008) and, importantly, the notion of probability tree models has until now not been formalised as presented in this thesis.

When representing a model by a probability tree, we often not only label edges by primitive probabilities but also by their meaning in a context of interest. For instance, in the coin-toss

4 These two vectors are equal up to a permutation of their components, so equal up to an interpretation of these labels in the context represented by the two graphs.

example in Fig. 0.1.1, edges are also labelled by 'heads' and 'tails'. In order to avoid ambiguity in the graphical representation of a tree model, this extra identification should always be recoverable from the graphical representation. We hence introduce some extra notation to ensure this is the case.

Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a probability tree and $\mathcal{T} = (V, E)$. Because we only consider finite graphs, we can always identify the set of root-to-leaf paths $\Lambda(\mathcal{T})$ of that tree with some discrete space Ω of the same cardinality. A bijection

$$\iota_\mathcal{T} : \Omega \to \Lambda(\mathcal{T}), \quad \omega \mapsto \Big(e \mid e \in E(\iota_\mathcal{T}(\omega))\Big) \tag{1.15}$$

which maps an element of that space to a sequence of edges is called a *tree embedding*. This map enables us to identify a graphical notion λ with an underlying model interpretation ω and vice versa. So again in the coin-toss example, we set up a model to describe events in the discrete space $\Omega = \{$(heads,heads), (heads,tails), (tails)$\}$. Every event $A \subseteq \Omega$ in that space can be depicted in the tree as a union of paths $\{\lambda \mid \lambda \in \iota_\mathcal{T}^{-1}(A)\}$. Events in Ω which have zero probability are not depicted in the tree graph. The existence of an edge $e = (v, v') \in E$ in a tree representation can then be interpreted as stating the possibility of the event $\iota_\mathcal{T}^{-1}(\Lambda(v))$ happening before $\iota_\mathcal{T}^{-1}(\Lambda(v'))$. So the path sigma-algebra of the event tree is in one-to-one correspondence with the sigma-algebra of events of an underlying problem description and induces a pre-order on the latter. A detailed discussion of this will be provided in Section 3.2 and Chapter 4 with extensive illustrations in Smith et al. (2017).

Importantly, a distribution $\pi_{\boldsymbol{\theta},\mathcal{T}}$ which factorises according to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ induces a probability measure $P_{\boldsymbol{\theta}} = \Pi_{\boldsymbol{\theta},\mathcal{T}} \circ \iota_\mathcal{T}$ on the underlying space Ω whose values can be calculated without explicitly referring to the tree graph. The tree model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$ is thus a parametric model on Ω. As a consequence, the probability space $(\Lambda(\mathcal{T}), \sigma(\mathcal{T}), \Pi_{\boldsymbol{\theta},\mathcal{T}})$ represented by any probability tree representation $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ of $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$ can be identified with the probability space $(\Omega, \sigma(\Omega), P_{\boldsymbol{\theta}})$. So we can deduce two properties of tree models here. First, if a problem is specified in terms of a relationship between events rather than random variables then these can be explicitly and transparently communicated using a tree embedding and an event tree representation of a given space as above. And second, we can characterise a tree model by imposing constraints on Ω or $P_{\boldsymbol{\theta}}$ without relying on a given graphical representation. Both of these observations will be central to our analyses in the subsequent chapters.

1.2.2. Bayesian networks as probability trees

Of course sometimes a problem is naturally defined through the relationships between a set of prespecified random variables. An example of this is when the model is a Bayesian network

defined via a collection of conditional independence assumptions on problem variables as in Example 1.3. When this is so, the semantics we develop below enable us to exploit the information coded in these variables using a probability tree model. Tree models thus contain Bayesian network models as a special case.

Consider a parametric model in the *positive discrete distribution framework* (Studený, 2005) where a discrete probability space $(\Omega, \sigma(\Omega), P)$ is equipped with a strictly positive measure. Here, we assume again that the measure $P = P_{\boldsymbol{\theta}}$ can be parametrised using $\boldsymbol{\theta}$ from a space $\Theta \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$. Let $X = (X_1, \ldots, X_m) : \Omega \to \mathbb{X}$ be a vector of discrete random variables on that space which are measurable with respect to the given measure and take values in a product state space $\mathbb{X} = \mathbb{X}_1 \times \ldots \times \mathbb{X}_m$, $m \in \mathbb{N}$. Suppose further that this probability measure can be written in the monomial form

$$P_{\boldsymbol{\theta}}(X = x) = \prod_{i=1}^{k} \theta(x_{A_i}) \text{ for all } x \in \mathbb{X} \tag{1.16}$$

where x_{A_i} denotes the vector $(x_j \mid j \in A_i) \in \mathbb{X}_{A_i} = \bigtimes_{j \in A_i} \mathbb{X}_j$ for index sets $A_i \subseteq \{1, \ldots, m\}$, $i = 1, \ldots, k$ and $k \in \mathbb{N}$: see Lauritzen (1996) for this notation.

Then the map $\Psi : \boldsymbol{\theta} \mapsto \boldsymbol{p}_{\boldsymbol{\theta}} = \left(p_{\boldsymbol{\theta}}(x) \mid x \in \mathbb{X}\right)$ which maps a choice of these parameters to an atomic probability $p_{\boldsymbol{\theta}}(x) = P_{\boldsymbol{\theta}}(X = x)$, $x \in \mathbb{X}$, is a monomial parametrisation. By construction, it thus defines a discrete parametric statistical model $\mathbb{P}_\Psi$ as in (1.2). This model captures assumptions on the problem variables implicitly, that is via the probability mass function, rather than explicitly in a graph.

Following Smith and Anderson (2008), we can now embed the state space $\mathbb{X}$—rather than Ω—into the set of paths of an event tree $\mathcal{T} = (V, E)$ via a tree embedding

$$\begin{aligned} \iota_{\mathcal{T}} = \iota_{\mathcal{T},A} : \quad & \mathbb{X}_1 \times \mathbb{X}_2 \times \ldots \times \mathbb{X}_m \to \Lambda(\mathcal{T}) \\ & (x_1, \ldots, x_m) \mapsto \left(e(x_{A_1}), \ldots, e(x_{A_k})\right) \end{aligned} \tag{1.17}$$

such that atomic probabilities are identified. So $\pi_{\boldsymbol{\theta},\mathcal{T}}(\iota_{\mathcal{T}}(x)) = \prod_{i=1}^{k} \theta(e(x_{A_i})) = P_{\boldsymbol{\theta}}(X = x)$ for all $x \in \mathbb{X}$. Because the random variables above have been assumed to have a strictly positive distribution, the resulting tree is then a probability tree. Of course this embedding into an event tree is also possible when certain (marginal) outcomes in the state space can be assigned probability zero. This will then translate into zero probabilities on edges: a case we excluded in Definition 1.5 in order to avoid issues associated with *faithfulness* of graphical models (Smith et al., 2017). An example of this was given in Görgen et al. (2015) and will be presented here in an extended form on page 26 below.

We assume in (1.17) that those index sets which are non-empty $A_i \neq \emptyset$ are pairwise different, $A_i \neq A_j$ for $i \neq j$. This is in order to be able to unambiguously associate one edge in the tree with one (marginal) outcome in the state space. For instance, in practice $A_i = \{1, 2, \ldots, i\}$ is often given by an index and the indices of all of its predecessors in the random vector: see below and Smith et al. (2017) for more details.

Definition 1.8 (X-compatible)**.** Let $(\mathcal{T}, \theta_{\mathcal{T}})$ be a probability tree whose set of root-to-leaf paths can be identified with the product state space $\mathbb{X}$ of a vector X of random variables as in (1.17). If in the corresponding tree embedding the set $A = \{A_1, \ldots, A_k\}$ of index sets is the same for every state $x \in \mathbb{X}$, so if every atom is embedded in the same order along every root-to-leaf path, we call this probability tree *X-compatible*.

In particular, every Bayesian network model on random variables X can now be represented by an X-compatible probability tree.

The monomial parametrisation in (1.16) is then often one of two types. First, whenever a recursive factorisation of a probability mass function according to an acyclic digraph is based on the *local Markov property*—stating that every vertex is independent of its *ancestor* vertices given its parents—then the θ-parameters in (1.16) are conditional probabilities of the type $p_i(x_i|x_{\mathrm{pa}(i)})$ for all $i = 1, \ldots, k$. Alternatively, in *decomposable* BN models the probability mass function (1.16) can take the more compact form

$$p_\theta(x) = \prod_{j=1}^{k} \theta(x_{C_j}) \quad \text{for all } x \in \mathbb{X} \tag{1.18}$$

where the C_i, $i = 1, \ldots, k$, are *cliques*—so maximally complete sets of vertices—of an underlying acyclic digraph $\mathcal{D}$ and $B_j = C_j \cap C^j$, where $C^j = \bigcup_{i=1}^{j-1} C_i$ for $j = 2, 3, \ldots, k$, are *separators* between these cliques. Then the model represented by $\mathcal{D}$ respects precisely the conditional independence assumptions $X_{Cj} \perp\!\!\!\perp X_{Cj-1\setminus Bj} \mid X_{Bj}$ for all $j = 2, 3, \ldots, k$. This parametrisation of decomposable models is a natural one to choose because there are no conditional independence constraints between variables within the same clique—so there is no need to specify a local Markov condition between these as above. Hence, inference is often made from a *junction tree* (Jensen and Jensen, 1994) instead of from $\mathcal{D}$, or in non-decomposable models from a *DAG of chain components* (Lauritzen and Richardson, 2002) where certain components of a bigger acyclic digraph are decomposable.

As a result, X-compatible trees allow for a very straightforward interpretation of their edge labels.

Remark 1.9 (Potentials). In X-compatible trees, the meaning of an edge $e(x_{A_i}) = (v, v')$ of 'passing from situation v to v'' can be stated more precisely as '$x_{A_i \setminus A^{i-1}}$ happened' given that

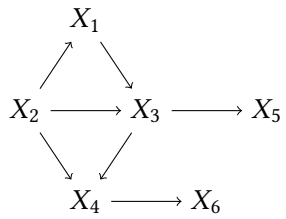


Figure 1.3. An acyclic digraph $\mathcal{D}$ depicting conditional independence assumptions on the random variables $X = (X_1, \ldots, X_6)$ which can be equivalently represented by the X-compatible tree constructed in Example 1.10.

'$x_{A^{i-1}}$ happened before', where $A^{i-1} = \bigcup_{j=1}^{i-1} A_j$ is the union of all indices in the monomial occurring before A_i and $i \geq 2$. This conditional interpretation is then the same along every root-to-leaf path because the order in which $\iota_{\mathcal{T}}$ embeds these states is the same for every atom. Thus, the primitive probabilities $\theta(x_{A_i}) = \theta(e(x_{A_i}))$ from (1.16) and (1.17) are *potentials* or *kernels* as in Lauritzen (1996). In particular, they have a conditional or marginal meaning that depends on the graph $\mathcal{T}$ and its sum-to-1 conditions in the parameter space $\Theta_{\mathcal{T}}$. As a consequence, the vectors of floret labels of an X-compatible tree are of the type $\boldsymbol{\theta}_v = \left(\theta(x_{A_i}) \mid x_{A_i \setminus A^{i-1}} \in \mathbb{X}_{A_i \setminus A^{i-1}}\right)$ for $v \in V$. They are thus rows of conditional probability tables of the underlying random variables.

Consider an illustration below.

Example 1.10 (Constructing an X-compatible tree)**.** Let $X = (X_1, X_2, \ldots, X_6)$ be a vector of random variables with joint probability mass function of the monomial form

$$p_{\boldsymbol{\theta}}(x) = \theta(x_{\{1,2,3\}})\theta(x_{\{2,3,4\}})\theta(x_{\{3,5\}})\theta(x_{\{4,6\}}) \tag{1.19}$$

for all $x \in \mathbb{X}_1 \times \ldots \times \mathbb{X}_6$ as in (1.18). Here, $p_{\boldsymbol{\theta}}$ is given in terms of a clique-based factorisation according to the decomposable acyclic digraph $\mathcal{D}$ in Fig. 1.3. Every indeterminate in the monomial in (1.19) is hence a potential as in Remark 1.9.

We can now draw an X-compatible tree using the embedding

$$\iota_{\mathcal{T}}(x) = \left(e(x_{\{1,2,3\}}), e(x_{\{2,3,4\}}), e(x_{\{3,5\}}), e(x_{\{4,6\}})\right) \tag{1.20}$$

for all $x \in \mathbb{X}$ as follows. The root-vertex will correspond to the joint random variable $X_{\{1,2,3\}} = (X_1, X_2, X_3)$ and every emanating edge $e(x_{\{1,2,3\}})$ will correspond to one element in the state space of these three variables, $(x_1, x_2, x_3) \in \mathbb{X}_1 \times \mathbb{X}_2 \times \mathbb{X}_3$. These edges are then labelled by their respective marginal probability $\theta(x_{\{1,2,3\}}) = p_{123}(x_1, x_2, x_3)$. Every child of the root corresponds to a *conditional* random variable. In particular, the vertex connected to the root

by the edge $e(x_{\{1,2,3\}})$ corresponds to the random variable $X_{\{2,3,4\}} \mid X_{\{1,2,3\}} = x_{\{1,2,3\}}$. Each of these vertices in turn has edges $e(x_{\{2,3,4\}})$ corresponding to states in the space associated to these variables, $x_{\{2,3,4\}} \in \mathbb{X}_2 \times \mathbb{X}_3 \times \mathbb{X}_4$. These edges are now labelled by the conditional probabilities $\theta(x_{\{2,3,4\}}) = p_4(x_4|x_2, x_3)$. Continuing in this way, we then attach children to these vertices which correspond to random variables $X_{\{3,5\}} \mid X_{\{1,2,3\}} = x_{\{1,2,3\}}, X_{\{2,3,4\}} = x_{\{2,3,4\}}$ and whose edges are labelled by $\theta(x_{\{3,5\}}) = p_5(x_5|x_3)$. To these we finally attach children and leaves corresponding to $X_{\{4,6\}} \mid X_{\{1,2,3\}} = x_{\{1,2,3\}}, X_{\{2,3,4\}} = x_{\{2,3,4\}}, X_{\{3,5\}} = x_{\{3,5\}}$ and whose edges are labelled by the conditional probabilities $\theta(x_{\{4,6\}}) = p_6(x_6|x_4)$.

The labelled event tree constructed here is now an X-compatible probability tree inducing a probability mass function $\pi_{\theta,\mathcal{T}} \circ \iota_{\mathcal{T}} = p_\theta$ which lies in the Bayesian network model $\{\boldsymbol{p}_{\boldsymbol{\theta}} \mid \boldsymbol{\theta} \in \Theta\}$ of distributions (1.19) which factor according to $\mathcal{D}$, where Θ is again a product of probability simplices—one for each row in the conditional probability tables of the components of X.

Interestingly, we could also have embedded these labels in a different order of vertices and edges: in fact, in any order compatible with $\mathcal{D}$. Each of the resulting X-compatible trees is then an alternative representation of the same Bayesian network model and the same tree model: see Example 3.29. Corollary 3.28 on page 93 will state this result in much more generality.

In the development in this section, a certain collection of random variables can be used to construct a labelled event tree which is a probability tree. We will hugely generalise this point in Chapters 2 and 3 where we state conditions under which a collection of monomials can be associated to a tree model.

We will now direct our focus on probability trees which, via an additional graphical property, can capture conditional independence assumptions on distributions which factorise according to their graph. We will then be able to provide expressive illustrations to the concepts introduced above, in particular in Example 1.12.

1.2.3. Staged trees

Probability trees are most interesting when two or more florets share the same labels, and distributions $\pi_{\theta,\mathcal{T}}$ factorise according to a 'coloured' graph $\mathcal{T}$ which captures these equalities. We will analyse this type of model in the remainder of this text.

We first present a definition adapted from Smith and Anderson (2008) and tailored to the development below.

Definition 1.11 (Staged tree)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with $\mathcal{T} = (V, E)$ and $\boldsymbol{\theta}_{\mathcal{T}} = (\boldsymbol{\theta}_v \mid v \in V)$ be a labelled event tree. We say that $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is a *staged tree* if all vectors of floret labels are either equal $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$ up to a permutation of their components or have disjoint sets of labels

$\{\theta(e) \mid e \in E(v)\} \cap \{\theta(e') \mid e' \in E(w)\} = \emptyset$ for any $v, w \in V$. We say that two vertices which have equal floret labels are in the same *stage* and we denote by $\sim$ the induced equivalence relation on the vertex set.

If no two related vertices lie on the same path, $\Lambda(v) \cap \Lambda(w) = \emptyset$ for any $v \sim w$, we will call the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ *square-free*.

For instance, the coin-toss model from the introduction (on page 3) is represented by a staged tree whose two inner vertices are in the same stage. This tree is not square-free.

An interpretation of stage structure is always based on the graph. In particular, given two vertices are in the same stage and a unit arrives at one of them, the transition probabilities to all children of that vertex will not depend on which of the two vertices the unit is actually in, and will thus not depend on the way that unit took to arrive in that situation. The edge (or transition) probabilities in these stages are in this sense independent of their history or location in the tree. See Thwaites and Smith (2015b) for a formal presentation of this type of conditional independence. Note that we will always colour all vertices in the same stage accordingly, as done in Barclay et al. (2013). In doing so, all assumptions on the distributions which factorise according to a staged tree can be coded in a purely graphical way.

When having a preassigned collection X of random variables as in Section 1.2.2 above, setting vectors of floret labels equal to each other in an X-compatible tree can be interpreted as specifying a set of *context-specific* conditional independences of the type $X_i \perp\!\!\!\perp X_j \mid X_k = x_k$ for some $i, j, k \in \{1, \ldots, m\}$. For instance, it is easy to see in Example 1.10 that many of the constructed vertices will be in the same stage. Indeed, often context-specific constraints hold only on subsets of the state spaces of a collection of random variables. These then provide structure which is additional to the one that can be represented in an acyclic digraph, and this structure is not of graphical nature. Models with these types of constraints are now widely used in BN modelling, especially when the domain of application is large (Boutilier et al., 1996; Smith, 2010).

Just like for Bayesian networks, stage constraints in staged tree models are by construction qualitative assumptions: whatever values two vectors of floret labels take, if their corresponding vertices are in the same stage then these labels will be identified. In the symbolic framework of later chapters, where we do not assign values to these indeterminates, we will thus often interpret the stage structure of a labelled event tree as a set of linear binomial constraints $\theta(e) - \theta(e') = 0$ on the edge labels, for $e \in E(v)$, $e' \in E(v')$ and $v \sim v'$.

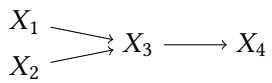
Note when a staged tree is square-free then no single edge label can appear twice on the same root-to-leaf path. As a consequence, the corresponding atomic monomials are then square-free (or multilinear). This constraint avoids various technical issues in non square-free trees

where the *global independence* assumption of parameters might be violated (Smith, 2010). It has recently been realised that in fact square-free classes of models provide very interesting objects of study in their own right (Thwaites and Smith, 2015b). In particular, all major results in Chapter 3 will apply only to the class of square-free staged trees. In Smith et al. (2017) we provide further insight into this class of models.

Consider now an example partly developed in Görgen et al. (2015) which illustrates the advantages of a staged tree representation over a Bayesian network model for the same problem.

Example 1.12 (Simplifying a BN using a staged tree)**.** We consider the following simplification of a real system described in Smith and Anderson (2008). A statistical model is designed to explain a possible unfolding of the following events in a cell culture. Initially, a cell finds itself in a benign or hostile environment. The level of activity between cells within this environment might be high or low, and if the environment is hostile then a cell gets damaged and might either survive or die. Surviving cells might make a full or partial recovery. We assume that the level of cell activity is independent of the environment being hostile or benign, whether or not a cell dies does not depend on its activity and that if it survives it will fully or partially recover with the same respective probabilities.

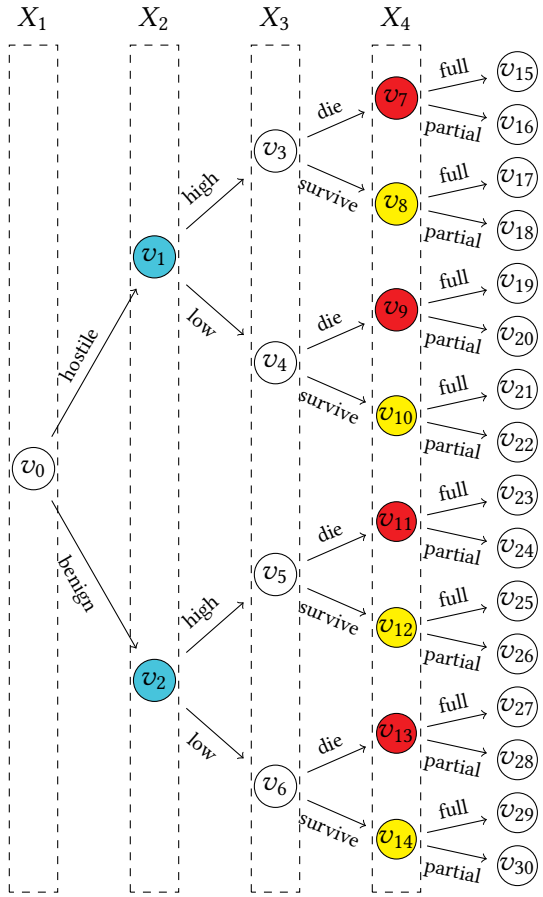
One ansatz is to model this situation using a Bayesian network on four binary random variables. The state of the environment is then represented by a random variable X_1 taking values in a state space $\mathbb{X}_1 = \{\text{hostile}, \text{benign}\}$, cell activity is measured by X_2 with $\mathbb{X}_2 = \{\text{high}, \text{low}\}$, viability via X_3 with $\mathbb{X}_3 = \{\text{die}, \text{survive}\}$ and recovery via X_4 with $\mathbb{X}_4 = \{\text{full}, \text{partial}\}$. Then our model assumptions translate into the conditional independence statements $X_1 \perp\!\!\!\perp X_2$ and $X_1, X_2 \perp\!\!\!\perp X_4 \mid X_3$. These can be represented by the acyclic digraph $\mathcal{D}$ with vertices X_1, X_2, X_3 and X_4, given below:



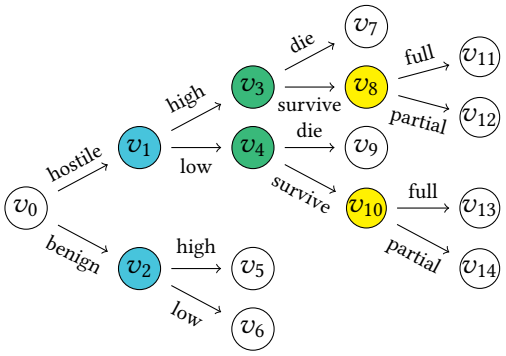
This graph represents the discrete Bayesian network model

$$\mathbb{P}_{\mathcal{D}} = \left\{ \boldsymbol{p} \in \Delta_{16-1} \mid \boldsymbol{p} = \left(p(x) \mid x \in \bigtimes_{i=1}^{4} \mathbb{X}_i \right),\ p \text{ factorises according to } \mathcal{D} \right\} \tag{1.21}$$

describing the above problem. First observe that $\mathbb{P}_{\mathcal{D}}$ and its representation $\mathcal{D}$ are not sufficient to accurately capture all assumptions on the $X_1, \ldots, X_4$ variables in the context we are interested in. In particular, the context-specific constraint that death or survival depends only on the state of the environment cannot be read from this graphical model. In addition, using a Bayesian network we retain redundant information in the product state space



(1.4.1) A 'degenerate' staged tree $(\mathcal{T}, \theta_{\mathcal{T}})_{\mathcal{D}}$ representation of the Bayesian network model $\mathbb{P}_{\mathcal{D}}$ from (1.21).



(1.4.2) A staged tree $(\mathcal{T}, \theta_{\mathcal{T}})$ accurately representing context-specific information.

Figure 1.4. Two alternative graphical representations for the biology model in Example 1.12.

$\mathbb{X} = \mathbb{X}_1 \times \mathbb{X}_2 \times \mathbb{X}_3 \times \mathbb{X}_4$. For instance, all states $(\text{benign}, x_2, x_3, x_4) \in \mathbb{X}$ have probability zero because there is no cell damage in a benign environment and all states $(x_1, x_2, \text{die}, x_3) \in \mathbb{X}$ are meaningless because if a unit has died, then there is no recovery possible.

Now, $\mathbb{P}_{\mathcal{D}}$ just like every discrete Bayesian networks admits an X-compatible staged tree representation. One possible such staged tree $(\mathcal{T}, \theta_{\mathcal{T}})_{\mathcal{D}}$ is given in Fig. 1.4.1. Note that because atoms which have probability zero in the BN need to have probability zero in the corresponding staged tree, $(\mathcal{T}, \theta_{\mathcal{T}})_{\mathcal{D}}$ is *degenerate* and includes edges which have zero probability. The apparent symmetries in this staged tree are typical for X-compatible staged trees with an underlying Bayesian network: all paths are of the same length and the stage structure depends on the distance of a vertex from the root. However, keeping in mind the assumptions made in our model, we notice that the bottom right part of the graph—in particular the root-to-leaf paths

ending in leaves $v_{23}, \ldots, v_{30}$—in Fig. 1.4.1 does not contain any valuable information. This is because this part of the graph depicts the redundant part of the product space identified above. $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\mathcal{D}}$ thus nicely illustrates how much superfluous information there is in the Bayesian network model $\mathbb{P}_{\mathcal{D}}$. Of course we could improve this representation using a context-specific Bayesian network rather than a Bayesian network but then the graphical nature of the model would be lost. So there is a strong case here for using a staged tree model instead[5].

We thus propose to model the situation at hand using a staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ as in Fig. 1.4.2. This new representation is not compatible with the four problem variables given above. It is however far more expressive than the acyclic digraph $\mathcal{D}$ and is less cluttered than the corresponding tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\mathcal{D}}$, whilst conveying *all* model information. In particular, all edges with probability zero and all unfoldings which are meaningless have been deleted, and the stage structure visually expresses the given conditional independence assumptions. We here identify the floret labels $\boldsymbol{\theta}_{v_1} = \boldsymbol{\theta}_{v_2}$—in a blue-coloured stage—because the activity between cells does not depend on the environment so that the transition probabilities from v_1 and v_2 are the same, $\boldsymbol{\theta}_{v_3} = \boldsymbol{\theta}_{v_4}$—in a green-coloured stage—because death or survival does not depend on cell acitivity given the environment is hostile, and $\boldsymbol{\theta}_{v_8} = \boldsymbol{\theta}_{v_{10}}$—in a yellow-coloured stage—because the chances of recovery of a surviving cell are independent of the history of that cell. The staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ then represents the model

$$\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})} = \left\{ \boldsymbol{\pi} \in \Delta^{\circ}_{8-1} \mid \boldsymbol{\pi} = \left(\pi_{\boldsymbol{\theta}, \mathcal{T}}(\lambda) \mid \lambda \in \Lambda(\mathcal{T}) \right),\ \pi_{\boldsymbol{\theta}, \mathcal{T}} \text{ factorises according to } \mathcal{T} \right\}. \quad (1.22)$$

Now, this new tree model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$ does not only have a simpler graphical representation and is a more apt representation of the situation at hand. It also lies in a much lower-dimensional probability simplex than the Bayesian network model $\mathbb{P}_{\mathcal{D}}$ from (1.21). In this type of *asymmetric* modelling context where a large amount of context-specific information is present, the use of staged tree models is thus highly advantageous.

The staging of a probability tree often respects certain symmetries as in the example above. In particular, in the development below we will make use of the following notion:

Definition 1.13 (Stratified trees)**.** Let $\mathcal{T} = (V, E)$ be an event tree. We say that a vertex $v \in V$ is at *level* i of $\mathcal{T}$ if the directed subpath from the root v_0 to v has i edges, $i \in \mathbb{N}$. We call a staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ *stratified* if all vertices which are in the same stage are also at the same level of the tree.

For example the staged trees in Fig. 1.4 are stratified.

5 Of course, if observational data was available then model selection techniques can help decide which model is a better fitting description of the problem at hand. See also numerous discussions of this type in Smith et al. (2017).

By Cowell and Smith (2014); Collazo and Smith (2015), the class of stratified staged tree models is amenable to various fast search algorithms. In particular, an X-compatible staged tree as in Definition 1.8 is stratified only if its stage constraints are of the form

$$\theta(x_A) = \theta(x'_A) \quad \text{for some } x_A, x'_A \in \mathbf{X}_A \tag{1.23}$$

where $A \subseteq \{1, \ldots, m\}$ in the notation of (1.16). For instance, the trees in Examples 1.3 and 1.10 have constraints of this type. So all X-compatible staged trees representing (context-specific) BN models are stratified and the conditional independence assumptions which are implicit in their probability distributions (1.16) and (1.18) can be straightforwardly read off an X-compatible graphical representation of such a model. Here, the stratification constraint (1.23) can be used to prevent an identification of primitive probabilities which do not belong to the same random variable and might not make sense in a modelling context. For instance, in Example 1.12 we might not want to identify the probabilities of high or low cell activity with the probabilities of a full or partial recovery of a cell.

By construction, stratified trees are also always square-free.

Before concluding this section, we now briefly present a second example of a stratified staged tree below which relates to Example 1.3 and which will provide some interesting illustrations later in the text.

Example 1.14 (Stratified trees and BNs)**.** The staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ depicted in Fig. 1.5 is a simplified detail of the one analysed in Barclay et al. (2013). Here, every atom in an underlying space is represented by a root-to-leaf path with two edges and corresponds to a possible history of a child in the study analysed by Fergusson et al. (1986). The first edge of each such path depicts the socio-economic background of a child, the second corresponds to a number of life events. For instance, $\lambda = ((v_0, v_2), (v_2, v_8)) \in \Lambda(\mathcal{T})$ represents a history of 'high social status, low economic background and low number of life events'. More detail on these measurement variables can be found in Section 3.2.4.

We can now embed information of the type 'once we know the social status of a child, then her number of life events is not further influenced by her economic situation'. In Fig. 1.5, the vertices v_1, v_2 and v_3, v_4 are then in the same stages which are coloured blue and green, respectively. So the primitive probabilities of the edges of the corresponding florets are identified, $\theta(v_i, v_j) = \theta(v_{i+1}, v_{j+2})$ for $j = 2i + 3, 2i + 4$ and $i = 1, 3$.

Let now S, E and L be three binary random variables with a strictly positive joint probability mass function

$$p_{\boldsymbol{\theta}}(s, e, l) = \theta(s, e)\theta(s, e, l) \tag{1.24}$$

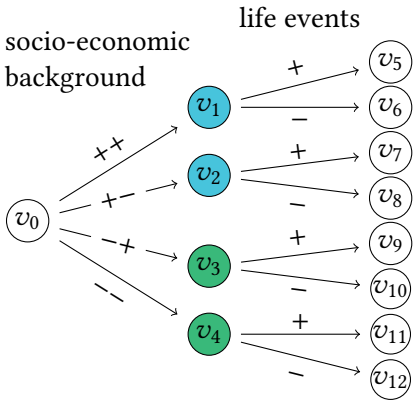


Figure 1.5. A stratified staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$, simplified version taken from Barclay et al. (2013). We label the edges by + and −, corresponding to 'high' and 'low', respectively. See Example 1.14 and Section 3.2.4 for a discussion.

for all $(s, e, l) \in \{\text{high}, \text{low}\}^3$. Here, S represents the social status of a child, E the economic background and L the number of life events. Then the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 1.5 is (S, E, L)-compatible and stratified. The staging of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is equivalent to the conditional independence assumption $E \perp\!\!\!\perp L \mid S$. Thus, Fig. 1.5 is a staged tree representation for the parametric model in Example 1.3 with $X_1 = S$, $X_2 = E$ and $X_3 = L$. In particular, for this problem

$$\theta(s, e, l) = \theta(s, e', l) \quad \text{for all } e \neq e' \tag{1.25}$$

is the stratification constraint as in (1.23). We will henceforth write $\theta(s, l) = \theta(s, e, l)$ for primitive probabilities belonging to vertices in those stages, acknowledging that these do not depend on the values taken by the random variable $E = e$. So simply $p_{\boldsymbol{\theta}}(s, e, l) = \theta(s, e)\theta(s, l)$ in (1.24).

1.2.4. Chain event graphs

Despite the apparent advantages of the staged tree model in Example 1.12 over an acyclic digraph representation for the same problem, we can also guess from the Fig. 1.4 that even in moderately sized problems event trees can easily become huge. This is the main motivation for Smith and Anderson (2008) to introduce a more compact graphical representation of the same model which is based on an event tree but avoids a tree's graphical redundancies. This object—called the *chain event graph*—has both representational and computational advantages whilst retaining the expressiveness of a staged tree. Chain event graphs are the main focus of the vast majority of publications in staged tree models and enable us to visualise more easily the structural implications of stages. However, for the focus of this thesis staged tree repres-

entations are more expressive and more natural links to the algebraic specification of this class of statistical models. So although not central to the development presented below, we finish this subsection with a brief description of this alternative graphical representation.

Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ denote a staged tree with $\mathcal{T} = (V, E)$. We then denote by $U_{\mathcal{T}} = V/\sim$ the partition on the vertex set induced by the stage relation $\sim$ on that tree. The set $U_{\mathcal{T}}$ is called the *stage set* of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. If a distribution factorises according to a coloured tree, its atomic probabilities are given by products of primitive probabilities along paths where labels are identified if two vertices are in the same stage. So the path sigma-algebra and the stage set of one representation are sufficient to identify a model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$. We will now translate that sigma-algebra and the stages into analogous objects in a different graph.

First note that stage structure, especially in stratified trees, is often in a sense symmetric along different paths. We will thus say that two vertices $v, v' \in u$ which are in the same stage $u \in U_{\mathcal{T}}$ are also in the same *position* if the induced probability subtrees $(\mathcal{T}(v), \boldsymbol{\theta}_{\mathcal{T}(v)})$ and $(\mathcal{T}(v'), \boldsymbol{\theta}_{\mathcal{T}(v')})$ represent the same model in the same parametrisations $\Psi_{\mathcal{T}(v)} = \Psi_{\mathcal{T}(v')}$[6]. In this case, the unfoldings of events from v and v' are the same and have the same attached probabilities up to the leaves of the tree. For instance, in Fig. 1.4.1 the vertices v_1 and v_2 are not in the same position but in Fig. 1.4.2 the vertices v_3 and v_4 are.

Now, the position relation induces a partition $W_{\mathcal{T}}$ on the vertex set V which is coarser than the one induced by the stage relation: every two vertices in the same position are trivially also in the same stage but the converse is not true. In particular, all leaves in V are in the same position, denoted $w_\infty \in W_{\mathcal{T}}$ and called a *sink* node, and the root is always the only element of a position denoted $w_0 \in W_{\mathcal{T}}$ and also called a *root*.

Definition 1.15 (Chain event graph)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree, $\mathcal{T} = (V, E)$. Denote the set of positions of this tree by $W_{\mathcal{T}}$. We construct a new labelled graph $(C(\mathcal{T}), \boldsymbol{\theta}_{\mathcal{T}})$ as follows:

$C(\mathcal{T}) = (W, F)$ is a graph with vertex set $W = W_{\mathcal{T}}$ given by the set of positions in the underlying staged tree. F is a set of possibly multiple edges between these vertices with the following properties. If there exist edges $e = (v, v'), e' = (w, w') \in E$ and v, w are in the same position then there exist corresponding edges $f, f' \in F$ emanating from a common vertex. If also v', w' are in the same position, then $f = f'$. The labels $\theta(f)$ of edges $f \in F$ in the new graph are inherited from the corresponding edges $e \in E$ in the staged tree.

We will henceforth call a labelled graph $(C(\mathcal{T}), \boldsymbol{\theta}_{\mathcal{T}})$ as above the *chain event graph (CEG)* of the underlying staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

Figure 1.6 shows a CEG whose stged tree was given in Fig. 1.4.2.

6 This implies that the labels of these two trees $\boldsymbol{\theta}_{\mathcal{T}(v)} = \boldsymbol{\theta}_{\mathcal{T}(v')}$ are equal up to a permutation. In the language of Chapter 3, these subtrees are 'polynomially equivalent': see Definition 3.12.

The CEG often provides a much more compact representation of a staged tree. This is because every structure of the tree graph that is repetitive (and in this sense redundant) is merged into a single vertex or edge. Colours in this new graph are retained only if they are not redundant, so only if they identify two vertices which are in the same stage but not in the same position. The primitive and atomic probabilities are hereby preserved, as are the identifications of root-to-leaf/sink paths and atoms in an underlying space. We thus find:

Proposition 1.16 (Smith and Anderson (2008))**.** *A staged tree* $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ *and its corresponding CEG* $(C(\mathcal{T}), \boldsymbol{\theta}_{\mathcal{T}})$ *are graphical representations of the same discrete statistical model.*

Consider now an example below where we construct the CEG for a staged tree which we have analysed in the previous section.

Example 1.17 (A CEG for Example 1.12)**.** Consider the staged tree model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$ from (1.22) and its representation $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ given in Fig. 1.4.2. The stage set $U_{\mathcal{T}}$ of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ contains precisely the following elements: the root $u_0 = \{v_0\}$, the blue-coloured stage $u_1 = \{v_1, v_2\}$, the green-coloured stage $u_2 = \{v_3, v_4\}$, the yellow-coloured stage $u_3 = \{v_8, v_{10}\}$ and all leaves in the uncoloured stage $u_\infty = \{v_5, v_6, v_7, v_{11}, v_{12}, v_{13}, v_{14}\}$. Note that the root is the only 'trivial' stage, containing a single vertex.

The positions in $W_{\mathcal{T}}$ are $w_0 = u_0$, $w_1 = \{v_1\}$, $w_2 = \{v_2\}$, $w_3 = u_2$, $w_4 = u_3$ and $w_\infty = u_\infty$. Thus, we can now construct the CEG $(C(\mathcal{T}), \boldsymbol{\theta}_{\mathcal{T}})$ corresponding to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with vertex set $W = W_{\mathcal{T}}$ and edges with labels inherited from $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ as in Definition 1.15. This is given in Fig. 1.6. Note here that v_1 and v_2 are the only vertices which are in the same stage but not in the same position, so w_1 and w_2 remain coloured blue. The remaining green and yellow colouring is redundant and hence omitted in the CEG.

We can read the underlying atoms from the set of root-to-sink paths of the CEG $(C(\mathcal{T}), \boldsymbol{\theta}_{\mathcal{T}})$ just like we could from $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. At the same time, the graph in Fig. 1.6 is less repetitive and has fewer vertices and edges than the one in Fig. 1.4.2.

Just like a staged tree, a CEG is a purely graphical representation of a model which is able to embody in its graph information about the underlying probability distribution. This information includes the number of atoms of the sample space, the levels of all associated random variables if the underlying staged tree is X-compatible, logical constraints on state spaces, equalities of marginal or conditional distributions and a certain order of events. It is easier to handle than a staged tree representation and yields a more compact problem description. Thanks to this graphical compactness, models represented by CEGs are open to a number of fast and efficient statistical inference techniques: model fit and learning in these models have been discussed in Freeman and Smith (2011); Collazo and Smith (2015), propagation algorithms

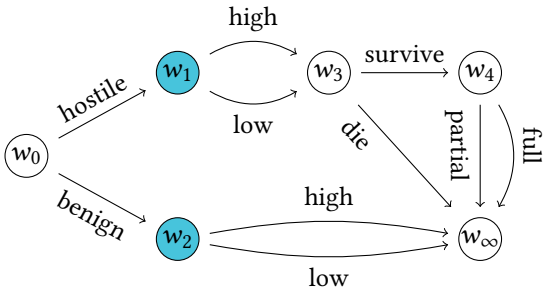


Figure 1.6. A CEG $(C(\mathcal{T}), \theta_{\mathcal{T}})$ whose underlying staged tree $(\mathcal{T}, \theta_{\mathcal{T}})$ is depicted in Fig. 1.4.2. See Examples 1.12 and 1.17.

were developed in Thwaites et al. (2008), seperation theorems are presented in Thwaites and Smith (2015b) and finally Thwaites et al. (2010); Thwaites (2013); Cowell and Smith (2014) laid the foundations for causal inference in CEGs which we will enhance later in this text. In addition, just like BN models, staged trees and CEGs can also be directly elicited by a domain expert (Smith, 2010; Smith et al., 2017). Recent developments have further extended the language of CEGs and staged trees to a decision theoretic domain (Thwaites and Smith, 2015a).

Importantly, the staged tree and its corresponding CEG can be used interchangeably to represent the same model. So all results mentioned above apply to both graphs and are always directly transferable. In terms of a formal analysis of the underlying model, we will exploit this transferability over the remainder of this text and usually base our results on properties of a staged tree. This is because these semantics are more easily translated into the algebraic framework below. So in the following, the main notions our analysis relies on are probability trees—or labelled event trees—together with their florets and induced subtrees, staged trees which are probability trees that include a number of identified florets, and stratified trees which are staged trees whose stage structure is self contained along the levels of the tree. In order to contrast our models to the more well-known Bayesian networks, we will also make use of the notion of X-compatible trees, so probability trees together with a set of problem variables which has been specified a priori.

1.3. Algebraic Statistics

Most of this work builds on the idea that we can analyse probability trees and staged trees as labelled graphs in a symbolic framework, where every edge label is an indeterminate that can potentially be assigned a meaning or a numerical value. Calculations involving these labels are then often of polynomial or rational form: see below. This interpretation opens the door

to employ techniques from algebra and algebraic geometry to inference in staged tree models. In this section we provide the necessary background for such an algebraic study.

In particular, throughout this thesis we will then focus on two notions. In Chapter 2, we will specify stage constraints and polynomial constraints on the—unknown—atomic probabilities in order to code all assumptions in a given staged tree model. Then in Chapter 3 we define a formal polynomial in the unknown edge labels which enables us to capture all information encoded in a labelled event tree. In this way we not only find that there is an intrinsic link between these polynomial notions and tree models but also that certain tools associated with algebraic geometry can be applied in a straightforward way in order to answer inferential queries in models represented by probability trees.

1.3.1. A very abbreviated introduction to algebraic geometry

We first recall some basic notions from Cox et al. (2015).

Let $S = \{s_1, \ldots, s_d\}$ denote a finite set of indeterminates with $d \in \mathbb{N}$ elements. A *(real) polynomial ring* $(\mathbb{R}_d[S], +, \cdot)$ is a commutative ring with an addition and a product whose elements are formal polynomials $f = \sum_{i=1}^{n} r_i s_1^{\alpha_{i,1}} s_2^{\alpha_{i,2}} \cdots s_d^{\alpha_{i,d}}$ in the given indeterminates, with real coefficients $r_i \in \mathbb{R}$ and non-negative exponents $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,d}) \in \mathbb{Z}^d_{\geq 0}$, for $i = 1, \ldots, n$ and $n \in \mathbb{N}$. Two polynomials are equal within this ring, $f = g \in \mathbb{R}_d[S]$, if and only if they coincide as polynomial functions, $f = g : \mathbb{R}^d \to \mathbb{R}$, when assigning values to the indeterminates in S. The map between an interpretation of a polynomial as an element in a ring structure and as a function is called the *evaluation homomorphism*. This homomorphism will be of use whenever we switch from a formal analysis of polynomials arising in inference in staged trees to an interpretation in a context where indeterminates are unknown probabilities.

The two rings we will be most interested in are $\mathbb{R}[\theta(e) \mid e \in E]$, the polynomial ring in indeterminates given by edge labels of a probability tree, and $\mathbb{R}[\pi_1, \ldots, \pi_n]$, the polynomial ring in indeterminates given by atomic probabilities. It is important to note that even though atomic probabilities are products of edge probabilities, $\pi_i = \prod_{e \in E(\lambda_i)} \theta(e)$, for $i = 1, \ldots, n$, the respective rings in these indeterminates are formally very different objects. We show below how quantities of interest can be calculated in both rings.

An *ideal* $I \subseteq \mathbb{R}_d[S]$ is a set of polynomials which is closed under outer multiplication, $f \cdot g \in I$ for all $f \in \mathbb{R}_d[S]$ and $g \in I$, and where $(I, +)$ is a subgroup of $(\mathbb{R}_d[S], +)$. In particular, the set

$$\langle g_1, \ldots, g_k \rangle = \left\{ \sum_{i=1}^{k} f_i g_i \mid f_i \in \mathbb{R}_d[S],\ i = 1, \ldots, k \right\} \subseteq \mathbb{R}_d[S] \tag{1.26}$$

defines an ideal *generated by* the polynomials $g_1, \ldots, g_k \in \mathbb{R}_d[S]$ for some $k \in \mathbb{N}$. For instance in Chapter 2, we will analyse properties of the ideal generated by a collection of stage constraints: compare also Example 1.3.

In this context we will frequently make use of two operations on ideals. The *sum*

$$I + J = \{f + g \mid f \in I, g \in J\} \tag{1.27}$$

of two ideals $I, J \subseteq \mathbb{R}_d[S]$ is the smallest ideal containing both I and J. Note that the union of two ideals—simply defined as set union—is not an ideal itself. So the union is not a well-defined operation on ideals and cannot be used to replace the sum above. However, if the ideals I and J are each generated by a set of polynomials[7] as in (1.26), then their sum is generated by the union of these sets: $\langle g_1, \ldots, g_k \rangle + \langle f_1, \ldots, f_l \rangle = \langle g_1, \ldots, g_k, f_1, \ldots, f_l \rangle$ where $l \in \mathbb{N}$.

The second operation we will employ is the intersection $\cap$ of ideals and is simply defined as set intersection.

The set of common zeros of all polynomials in an ideal

$$\mathbf{V}(I) = \left\{x \in \mathbb{R}^d \mid g(x) = 0 \text{ for all } g \in I\right\} \tag{1.28}$$

is called a *variety*, and sometimes an algebraic variety. We use the bold operator symbol $\mathbf{V}$ to distinguish these sets from vertex sets V. Varieties are solution sets of systems of polynomial equations. They will enable us to specify sets of points which fulfil equations coding model assumptions. We often use that if one ideal is contained in another $I \subseteq J$ then its variety contains the variety induced by the other ideal, $\mathbf{V}(J) \subseteq \mathbf{V}(I)$. Over an algebraically closed field where every polynomial equation has a solution, much more precise statements concerning this duality of ideals and varieties can be made: see Chapter 4 of Cox et al. (2015).

For use in statistical inference it is central to note that varieties can be generalised to so-called *semi-algebraic sets* which are solution sets of polynomial equations and inequalities:

$$\mathbf{S}(F, G) = \left\{x \in \mathbb{R}^d \mid f(x) > 0 \text{ and } g(x) = 0 \text{ for all } f \in F, g \in G\right\} \tag{1.29}$$

where $F, G \subseteq \mathbb{R}_d[S]$ are collections of polynomials. Semi-algebraic sets are much harder to analyse than varieties but provide the correct framework to capture the behaviour of various classes of statistical models. This is because statistical models often include positivity constraints (or at least a specification of probabilities) which cannot be coded in a variety. For instance, probability simplices are semi-algebraic sets: see (2.25) below.

7 By Hilbert's Basis Theorem, every ideal is generated by a finite set of polynomials: see Theorem 4 in Section 2.5 of Cox et al. (2015).

Lastly, we will sometimes make use of an equivalence relation $\sim$ defined on a polynomial ring $\mathbb{R}_d[S]$ which relates two polynomials $g \sim h$ if and only if their difference lies in a certain ideal I in that ring, so $g - h \in I$. This relation induces a ring of equivalence classes $\mathbb{R}_d[S]/_\sim = \mathbb{R}_d[S]/I$, called a *factor ring*. Then the ideal $I \subseteq \mathbb{R}_d[S]$ is the class of all zeros in the factor ring $\mathbb{R}_d[S]/I$. One of the central results on these rings is the *isomorphism theorem* which states that every factor ring is isomorphic to a ring whose indeterminates are given by exactly one representative for each equivalence class: see Exercise 16 in Section 5.2 of Cox et al. (2015). This result will enable us to formally identify edge labels and impose sum-to-1 conditions below.

Algebraic geometry provides us with the tools to analyse ideals and varieties and the very close relationship between these two notions. Because it is not within the scope of this work to go too deeply into this intriguing field of mathematics, we will limit our observations here to one main point: it is often much easier to characterise properties of a collection of polynomials than it is to describe a set of points. So we will almost exclusively work on ideals rather than on varieties. In this study, we can then also use the freely available computer algebra software `CoCoA` (Abbott et al., 2016) for performing our computations.

The properties of ideals and varieties have already been widely and successfully used to capture and exploit the structure of statistical models. An algebraic approach to statistical inference has first been successfully formalised in the seminal works of Pistone et al. (2001a) and Pachter and Sturmfels (2005). Over the past ten years, the research area of *Algebraic Statistics* has then rapidly expanded from the initial use in discrete models, design of experiment (Maruri-Aguilar et al., 2013) and phylogenetics to a successful geometric characterisation of well-known graphical models (Garcia et al., 2005; Geiger et al., 2006), applications in Gaussian or log-linear models (Drton et al., 2009) and latent tree models (Settimi and Smith, 2000; Zwiernik, 2016) and other models with hidden variables (Mond et al., 2003), as well as a number of other fruitful applications in various domains of statistics (Gibilisco et al., 2010).

For instance, in the study of Bayesian networks, Geiger et al. (2006) found that the varieties which characterise conditional independence assumptions in decomposable acyclic digraphs are *toric*: they are solution sets of binomial equations of the form

$$\pi_1^{\alpha_{i,1}} \pi_2^{\alpha_{i,2}} \cdots \pi_n^{\alpha_{i,n}} = \pi_1^{\beta_{j,1}} \pi_2^{\beta_{j,2}} \cdots \pi_n^{\beta_{j,n}} \qquad \alpha_i, \beta_j \in \mathbb{Z}^n_{\geq 0} \tag{1.30}$$

in a ring of atomic probabilities $\mathbb{R}[\pi_1, \ldots, \pi_n]$. Toric ideals can be equivalently defined as elimination ideals (see Lemma 2.11) of kernels of monomial maps like (1.4): see Sturmfels (1996) for a full development of this theory. The result of Geiger et al. (2006) is repeated in Proposition 2.10 below. A similar result was obtained by Pistone et al. (2006) in a more general setting.

So very special types of graphical and statistical models can give rise to very special types of polynomial structures. We will elaborate on this point below.

1.3.2. Algebraic notions for staged trees

We will now present a short dictionary linking properties of staged trees to the notions from algebra introduced above, before going into much more detail in Chapter 2.

Let $\mathbb{P}_\Psi$ in the following always be a discrete and parametric statistical model with a monomial parametrisation $\Psi : \Theta \to \mathbb{P}_\Psi$. Then all elements in this model are vectors which can be written in the form $\left(\theta_1^{\alpha_{i,1}} \cdots \theta_d^{\alpha_{i,d}} \mid \alpha_i \in \mathbb{Z}^d_{\geq 0}, i = 1, \ldots, n\right)$ as in Definition 1.2. In an algebraic framework, we will again call the atomic probabilities in that vector atomic monomials: compare (1.12). By abuse of notation, we now always let $\mathbb{R}[\Theta] = \mathbb{R}[\theta_1, \ldots, \theta_d]$ denote the real polynomial ring in all indeterminates in a model parametrised by Ψ as above. In tree models $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})}$, that ring is also denoted $\mathbb{R}[\Theta_\mathcal{T}] = \mathbb{R}[\theta(e) \mid e \in E]$ and has indeterminates given by edge labels. So this ring is always particular to one parametrisation Ψ or one graphical representation $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$, $\mathcal{T} = (V, E)$, of a given model.

In Sections 1.1 and 1.2, we provided a probabilistic interpretation of the colouring in a staged tree, in terms of a collection of conditional independence assumptions between events. We can now express this notion in terms of ideals and polynomial rings.

Let first $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{\text{sat}} = (\mathcal{T}_{\text{sat}}, \boldsymbol{\theta}_{\mathcal{T}_{\text{sat}}})$ denote a trivially staged trees, whose vertices are all in stages which contain only that vertex itself. So here all edge labels are different. Because a model represented by such a tree equals[8] the whole probability simplex, $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{\text{sat}}} = \Delta^\circ_{n-1}$, we call its representation $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{\text{sat}}$ a *saturated* tree. If we impose stage structure on a saturated tree, then $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{\text{sat}}$ becomes a staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ with the same graph $\mathcal{T}_{\text{sat}} = \mathcal{T} = (V, E)$ where certain floret labels $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$ are now identified, $v, w \in V$. But this simply implies that the vector of all labels $\boldsymbol{\theta}_\mathcal{T}$ in the staged tree is the result of a projection of the labels $\boldsymbol{\theta}_{\mathcal{T}_{\text{sat}}}$ of the saturated tree. This is a projection from the parameter space of the saturated model $\bigtimes_{v \in V} \Delta^\circ_{\#E(v)-1}$ onto the parameter space of the staged tree model $\bigtimes_{u \in U_\mathcal{T}} \Delta^\circ_{\#E(u)-1}$, where $E(u)$ denotes the edge set of one representative of the stage $u \in U_\mathcal{T}$. So imposing stage structure on a probability tree is equivalent to applying a projection map on its parameter space. In particular, whenever new stage structure is imposed on a staged tree—by specifying additional pairs of vertices in the same stage—then we obtain a submodel of the original model whose parameter space is again a projection of the formerly specified space: see also Corollary 2.9 below.

8 See Proposition A.1 in the appendix.

Algebraically, the projection linking $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_{\text{sat}}$ to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is a ring homomorphism

$$\begin{aligned} \Phi_\mathcal{T} : \quad & \mathbb{R}[\Theta_{\mathcal{T}_{\text{sat}}}] \to \mathbb{R}[\Theta_\mathcal{T}], \\ & \theta(e) \mapsto \theta(e') \text{ whenever } e \in E(v),\ e' \in E(u) \text{ and } v \in u \in U \end{aligned} \tag{1.31}$$

where $U = U_\mathcal{T}$ is the set of stages of the staged tree.

The kernel $\ker(\Phi_\mathcal{T}) = \{f \in \mathbb{R}[\Theta_{\mathcal{T}_{\text{sat}}}] \mid \Phi_\mathcal{T}(f) = 0\}$ of that map is an ideal

$$I_U = \langle \theta(e) - \theta(e') \mid e \in E(v),\ e' \in E(u) \text{ and } v \in u \in U \rangle. \tag{1.32}$$

The degree-1 binomials generating the kernel I_U now capture the componentwise equations $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$ whenever $v, w \in u$ are in the same stage. We call I_U the *stage ideal* of $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. The isomorphism theorem then implies that the factor ring $\mathbb{R}[\Theta_{\mathcal{T}_{\text{sat}}}]/I_U$ whose zero element is the stage ideal is isomorphic to the polynomial ring $\mathbb{R}[\Theta_\mathcal{T}]$ of the staged tree. In simple words, this result can be interpreted as saying that first drawing a saturated tree and then embedding stage information yields 'the same' formal structure as the one obtained by directly drawing a staged tree.

The variety induced by the stage ideal is given by

$$\mathbf{V}(I_U) = \{\theta(e) = \theta(e') \mid e \in E(v),\ e' \in E(u) \text{ and } v \in u \in U\}. \tag{1.33}$$

So $\mathbf{V}(I_U)$ equals the space spanned by all indeterminates $\boldsymbol{\theta}_\mathcal{T}$ in the staged tree, as a subspace of the one spanned by all indeterminates $\boldsymbol{\theta}_{\mathcal{T}_{\text{sat}}}$ of the saturated tree. In other words, the parameter set of a staged tree can be identified from its saturated version together with the stage ideal. So stage ideals can be used to capture the assumptions made in a staged tree model.

Another constraint on staged trees—or in fact on all probability trees—requires that labels attached to the same floret must sum to unity. So in a fashion similar to the above we can define an ideal

$$I_1 = \langle \sum_{e \in E(v)} \theta(e) - 1 \mid v \in V \rangle \tag{1.34}$$

generated by these local sum-to-1 conditions on the labels of an event tree $\mathcal{T} = (V, E)$. Then again the factor ring $\mathbb{R}[\Theta_\mathcal{T}]/I_1$ whose zero element is given by the ideal above is isomorphic to a formal ring where these conditions hold.

Example 1.18 (Example 1.14 continued)**.** Return to the stratified tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ in Fig. 1.5 which we have analysed in the previous section. From (1.23), the stage ideal of this staged tree equals

$$I_U = \langle \theta(s,e,l) - \theta(s,e',l) \mid e \neq e', s, l = 0,1 \rangle \tag{1.35}$$

as a subset of the polynomial ring $\mathbb{R}[\Theta_{\mathcal{T}_{\text{sat}}}] = \mathbb{R}_{12}[\theta(s,e), \theta(s,e,l) \mid s,e,l = 0,1]$ of the saturated tree. As above, the factor ring $\mathbb{R}[\Theta_{\mathcal{T}_{\text{sat}}}]/I_{\mathcal{T}}$ is then isomorphic to the polynomial ring of the staged tree, $\mathbb{R}[\Theta_{\mathcal{T}}] = \mathbb{R}_8[\theta(s,e), \theta(s,l) \mid s,l = 0,1]$. So here the isomorphism theorem simply substitutes the labels $\theta(s,e,l)$ by $\theta(s,l)$, acknowledging that these conditional probabilities do not depend on the value e of the random variable E. This is simply an algebraic expression for the conditional independence assumption $E \perp\!\!\!\perp L \mid S$ we made in the example above.

In Geiger et al. (2006), the authors algebraically characterise the conditional independence assumption in this model using cross-product differences as in Example 1.3. So the model constraints are captured in a toric ideal

$$J = \langle p_{000}p_{101} - p_{001}p_{100},\ p_{010}p_{111} - p_{011}p_{110} \rangle \tag{1.36}$$

in a polynomial ring $\mathbb{R}[\boldsymbol{p}] = \mathbb{R}_8[p_{sel} \mid s,e,l = 0,1]$ whose indeterminates are the atomic probabilities $p_{sel} = p_{\boldsymbol{\theta}}(s,e,l)$ for $s,e,l \in \{0,1\}$, from (1.24). These constraints are the same for every acyclic digraph representation of the model and do not arise from stage structure as in our staged tree model. In fact, the toric variety $\mathbf{V}(J)$ intersected with the probability simplex is the set of all distributions over eight atoms for which the model assumptions are true:

$$\mathbb{P} = \mathbf{V}(J) \cap \Delta^{\circ}_{8-1} \tag{1.37}$$

precisely as in (1.7).

It is important to observe that here there is no analogous result for the stage ideal: the staged tree model does not equal the stage constraints plus imposed sum-to-1 conditions, so $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} \neq \mathbf{V}(I_U) \cap \mathbf{V}(I_1)$. This is because of the above mentioned parametrisation-dependence of the stage ideal. In particular, without a translation into atomic probabilities, the constraints on a staged tree are not sufficient to characterise the underlying model algebraically. We will present two new and alternative characterisations in Theorem 1 and Theorem 2 below.

Observe further that both $I_{\mathcal{T}}$ and $\mathbf{V}(I_{\mathcal{T}})$ have linear structure and are geometrically much simpler objects than the ideal J and toric variety $\mathbf{V}(J)$. A reparametrisation between the two characterisations of the model is given by the ring homomorphism $\Phi : \mathbb{R}[\boldsymbol{p}] \to \mathbb{R}[\Theta_{\mathcal{T}}]$ which maps an atomic probability to a product of conditional probabilities, $p_{sel} \mapsto \theta(s,e)\theta(s,l)$. The inverse of this map is rational, calculated using the law of total probability:

$$\Phi^{-1}: \quad \theta(s,e) \mapsto p_{se0} + p_{se1},\ \ \theta(s,l) \mapsto \frac{p_{s0l} + p_{s1l}}{p_{s00} + p_{s01} + p_{s10} + p_{s11}}. \tag{1.38}$$

Thus, whilst Φ is an invertible function and both ideals capture in this sense the same information, Φ is not a polynomial ring isomorphism, and both algebraic characterisations are very different.

The example above provides a first illustration of both the high potential of an algebraic characterisation of statistical models and the technical difficulty in using means from algebraic geometry in a non-algebra context. On the one hand, results like (1.36) and (1.37) enable us to employ computer algebra software to determine a statistical model as the solution set of a system of polynomial equations: we will give an in-depth presentation of this observation in Chapter 2. On the other hand, a naïve use of these techniques will almost always cause problems arising from positivity constraints on atomic probabilities, arising from sum-to-1 conditions or arising from the fact that in statistical inference we usually work over the real numbers. For instance, equivalence between the cross-product differences in (1.6) and (1.36) and the stage constraints in (1.8) and (1.35) is easy to see using a rational invertible map (1.38) as in the example above. However, this map is only applicable knowing that we divide by non-zero terms and that we can cancel out factors which appear in both numerator and denominator of these fractions. This is not immediately obvious in the framework introduced above and requires the introduction of the notion of a *quotient field*—a field of rational functions or fractions—and the notion of the *closure* of a semi-algebraic set in terms of a variety.

In short, doing algebraic geometry over a probability simplex Δ°_{n-1} rather than in Euclidian space $\mathbb{R}^d$ or in a complex space $\mathbb{C}^d$ is very hard and requires a large number of caveats. The various technical difficulties involved in such an analysis are not the focus of this thesis. However, because this field is highly intriguing and staged tree models have so far not been linked to results in algebraic geometry and algebraic statistics, we will take the following minimalistic approach in exploring this new ground. Within this text we will keep the algebraic notation to a minimum and we will foremostly work on the main notions (ideals and varieties) introduced above. Where subtleties which are not covered by this approach come into play, we refer to relevant results and suggest further reading in footnotes. Over the text, we will then provide extensive illustrations to the concepts we touch on rather than going into much theoretical detail.

Throughout the next three chapters, the following notions will thus be important: *staged trees*, often treated as labelled event trees whose floret labels might be identified without specifying concrete values, *tree parametrisations* which map these edge labels into probability simplices, *ideals* in real polynomial rings and *varieties* as well as *semi-algebraic sets* which capture the properties of the images of these mappings, and finally *polynomials* defined on edge labels

which can both be used as surrogates for labelled event trees and which can additionally be applied in a non-symbolic framework for calculating probabilities in staged tree models.

2. A geometric analysis of staged tree models

In this chapter, we will investigate a geometric characterisation of staged tree models. Our primary aim in this analysis is to provide a few insightful illustrations in order to answer the question: what do staged tree models look like as sets inside probability simplices?

We derive a specification of these sets in Theorem 1 in the first section below and then provide numerous illustrations over the remainder of the chapter. By definition, our sets have monomial parametrisations but are not always one-dimensional curves as depicted for example in Fig. 0.1.1. More generally, a set of distributions which factorise according to a staged tree is a semi-algebraic set as in Figs. 0.2 and 1.1. So our secondary goal is thus to show how tools which are well-known in (computational) algebra can be used to characterise staged tree models. In order to do this, we will translate properties of staged trees from the language of graphical models to the language of algebraic geometry. Because every Bayesian network is also a staged tree model and because this algebro-geometric approach has been well studied for BNs, we can hereby easily check that our results are in line with those in the current literature. The central step in this development is a translation of linear stage constraints on the edge labels of a staged tree into polynomial constraints on its atomic probabilities. In probabilistic terms, this translation is based on Bayes' rule: transforming an identification of conditional probabilities into an identification of functions of joint probabilities. In algebraic terms, the problem of translating an explicit parametrisation of a variety into an implicit characterisation as the solution set of a system of polynomial equations can be achieved using elimination theory, a generalisation of Gaussian elimination in linear algebra. We will show how calculations in both frameworks are deeply interlinked. In this way we will also be able to derive that the equations which define staged tree models can be given a straightforward interpretation in statistical inference: in particular, we prove that they are equations of odds ratios. Finally, we employ the developed techniques to a brief analysis of all staged tree models on three or four atoms and draw these as varieties which lie inside probability simplices.

2.1. A characterisation in terms of odds ratios

Our first aim is to deduce a characterisation of staged tree models which is not parametric, so a characterisation which is not specific to a particular choice of coloured tree representation of a family of probability distributions. In this development, for simplicity we start off by interpreting labels of a staged tree as primitive probabilities, so positive numbers strictly between zero and one.

By Definition 1.7, every distribution which factorises according to a given staged tree can be parametrised as a vector of monomials whose components are products of edge labels in that graph. Stage structure—normally represented by coloured vertices—is implicit in this monomial parametrisation, in the sense that some of the labels or indeterminates are identified with each other. So a characterisation of staged tree models as points inside a probability simplex must take these implicit constraints into account:

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \Delta^{\circ}_{n-1} \cap \{\boldsymbol{p} \in \mathbb{R}^n \mid f(\boldsymbol{p}) = 0 \text{ for all } f \in F\}. \tag{2.1}$$

The main question we are now interested in is: *What are these constraining functions F?* And is there a straightforward way to infer them from a collection of atomic monomials as above?

To illustrate this query consider again Fig. 1.1 on page 11. Up to this point, we have introduced stage constraints as assumptions on a certain parametrisation of a model. This approach requires that we are given a graphical representation $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ of the model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$ so that we can specify which vertices v and w are in the same stage and have their floret labels identified, $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$. These constraints then characterise the parameter space on the left hand side of that figure. Explicitly, they constrain a high-dimensional Euclidean space to a product of probability simplices, $\bigtimes_{v\in V} \Delta^{\circ}_{\#E(v)-1} \subseteq \mathbb{R}^d$. But what if we want to characterise the model itself, so the image of that space under a monomial parametrisation? In order to do this, we need to find a way of translating this stage structure into, preferably polynomial, constraints on the *atomic* probabilities. We can then characterise the set on the right hand side of our figure—so points in the model itself—independent of a parametrisation.

Advantages and disadvantages of both characterisations of a staged tree model are discussed in the next two sections.

In order to understand how stage constraints are transformed under a monomial parametrisation, observe first that primitive probabilities in staged trees can be easily expressed in terms of atomic probabilities.

Remark 2.1 (Primitive and atomic probabilities). Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree with graph $\mathcal{T} = (V, E)$, and let $\theta(e)$ be the label of an edge $e = (v, v') \in E$. Denote again by $\Pi_{\boldsymbol{\theta},\mathcal{T}}$ the probability

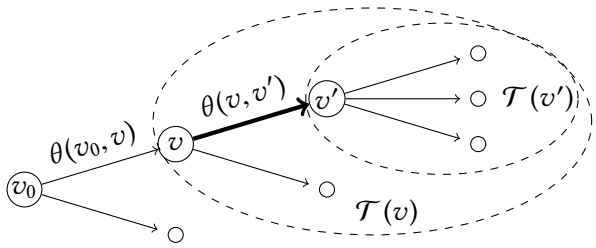


Figure 2.1. Primitive probabilities are fractions of probabilities of vertex-centred events. Here, the probability of $\Lambda(v)$ equals $\theta(v_0, v)$ and the probability of $\Lambda(v')$ equals $\theta(v_0, v)\theta(v, v')$ because the atomic probabilities in the induced subtrees $\mathcal{T}(v)$ and $\mathcal{T}(v')$ sum to one. So the primitive probability $\theta(v, v') = \frac{\theta(v_0, v)\theta(v, v')}{\theta(v_0, v)}$ can be written as in (2.2), and is in fact a conditional probability. See Remark 2.1.

measure defined by a staged tree as in Proposition 1.6. Then the primitive probability

$$\theta(e) = \frac{\Pi_{\boldsymbol{\theta}, \mathcal{T}}(\Lambda(v'))}{\Pi_{\boldsymbol{\theta}, \mathcal{T}}(\Lambda(v))} \tag{2.2}$$

is the conditional probability of passing through v' given that a unit has arrived at $v \in V$.

This is because in the fraction above both probabilities of events can be written as the probability of arriving at their respective central vertex multiplied by the sum of probabilities of all root-to-leaf paths of the induced subtree rooted at that vertex—and these sum to one. In formulae,

$$\Pi_{\boldsymbol{\theta}, \mathcal{T}}(\Lambda(v')) = \prod_{e' \in E(v_0 \to v)} \theta(e') \cdot \theta(v, v') \cdot \sum_{\lambda' \in \Lambda(\mathcal{T}(v'))} \pi_{\boldsymbol{\theta}, \mathcal{T}(v')}(\lambda') \tag{2.3.1}$$

$$\Pi_{\boldsymbol{\theta}, \mathcal{T}}(\Lambda(v)) = \prod_{e' \in E(v_0 \to v)} \theta(e') \cdot \sum_{\lambda \in \Lambda(\mathcal{T}(v))} \pi_{\boldsymbol{\theta}, \mathcal{T}(v)}(\lambda) \tag{2.3.2}$$

where we use the shorthand $E(v_0 \to v) \subseteq E$ to denote the set of edges in the path from v_0 to v in $\mathcal{T}$. By construction, $\sum_{\lambda' \in \Lambda(\mathcal{T}(v'))} \pi_{\boldsymbol{\theta}, \mathcal{T}(v')}(\lambda') = \sum_{\lambda \in \Lambda(\mathcal{T}(v))} \pi_{\boldsymbol{\theta}, \mathcal{T}(v)}(\lambda) = 1$. As a consequence, in the fraction (2.2) all monomials except for $\theta(e)$ cancel out.

See Fig. 2.1 for an illustration of this result.

Now, the expression derived in Remark 2.1 can be easily used to translate stage identifications into equations involving atomic probabilities. We will show how to do this below.

Throughout the development presented in this chapter, we will simplify notation as follows. Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree, $\mathcal{T} = (V, E)$, and let two vertices $v, w \in V$ be in the same stage. Without loss of generality we assume that these vertices are parents of leaves. We denote for simplicity the atoms in $\Lambda(v) = \{\lambda_1, \ldots, \lambda_k\}$ and in $\Lambda(w) = \{\lambda_{\Phi(1)}, \lambda_{\Phi(2)}, \ldots, \lambda_{\Phi(k)}\}$ where Φ is

the map which identifies those edges in the two associated florets which have the same colour: so that λ_i and $\lambda_{\Phi(i)}$ are root-to-leaf paths whose respective final edges have the same label, $i = 1, \ldots, k$ and $k = \#E(v)$. Henceforth, we will then also always write $\pi_i = \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda_i)$ for atomic probabilities, $i = 1, \ldots, n$. Consider Fig. 2.2 for an illustration.

Lemma 2.2 (Characterising stages)**.** *In the notation above, for v, w parents of leaves, the stage constraint $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$ is equivalent to the equations*

$$\pi_i \sum_{s=1}^{k} \pi_{\Phi(s)} = \pi_{\Phi(i)} \sum_{t=1}^{k} \pi_t \quad \text{for all } i = 1, \ldots, k. \tag{2.4}$$

Proof. Whenever two vertices v and $w \in V$ are in the same stage then by definition the two vectors of floret labels are identified, $\boldsymbol{\theta}_v = \boldsymbol{\theta}_w$. As a consequence, to every stage there is associated a system of equations $\theta(v, v') = \theta(w, w')$ for fitting children $v' \in \mathrm{ch}(v)$ and $w' \in \mathrm{ch}(w)$ of these vertices. Substituting the fraction (2.2) from Remark 2.1 into these stage equations, we obtain thus that v and w are in the same stage if and only if for all v', w' as above the following equation is true:

$$\frac{\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v'))}{\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v))} = \frac{\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(w'))}{\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(w))} \tag{2.5}$$

or equivalently,

$$\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v'))\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(w)) = \Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(w'))\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v)) \tag{2.6}$$

because atomic probabilities are assumed to be strictly positive.

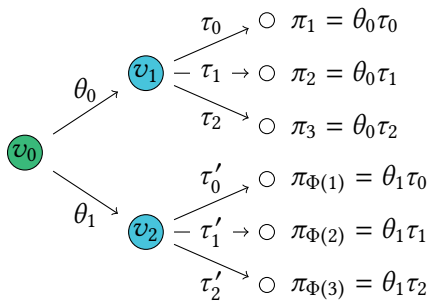
If v and w are vertices as above then the events centred at these vertices contain precisely the root-to-leaf paths ending in the children sets $\mathrm{ch}(v)$ and $\mathrm{ch}(w)$. So $\Lambda(v) = \bigcup_{v' \in \mathrm{ch}(v)} \Lambda(v') = \bigcup_{v' \in \mathrm{ch}(v)} \{\lambda_{v'}\}$ where $\lambda_{v'}$ denotes the only element of the event $\Lambda(v')$, for all $v' \in \mathrm{ch}(v)$. Hence,

$$\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v)) = \sum_{v' \in \mathrm{ch}(v)} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda_{v'}) = \sum_{t=1}^{k} \pi_t \tag{2.7}$$

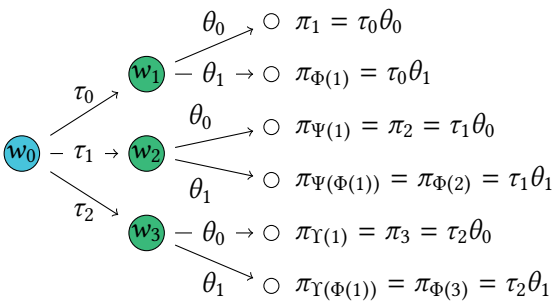
and in analogy for $\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(w))$ with v replaced by w and t replaced by $\Phi(t)$ in (2.7). The claim follows by substituting (2.7) into (2.6). □

Observe that if v and w are not parents of leaves, the atomic probabilities in (2.4) can be straightforwardly replaced by sums of probabilities in the corresponding vertex-centred events, and the claim remains true.

Consider a simple illustration below.



(2.2.1) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ whose stages are on leaf-florets, so $\tau_i = \tau_i'$ for $i = 1, 2, 3$.



(2.2.2) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ which represents the same model as $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ from Fig. 2.2.1.

Figure 2.2. Two probability trees whose stage structure is captured in form of polynomial equations in Examples 2.3 and 2.8.

Example 2.3 (Stage constraints in polynomial form)**.** In the probability tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ depicted in Fig. 2.2.1 let the vertices v_1 and v_2 have floret labels $\boldsymbol{\theta}_{v_1} = (\tau_1, \tau_2, \tau_3)$ and $\boldsymbol{\theta}_{v_2} = (\tau_1', \tau_2', \tau_3')$, respectively. We deduce from Remark 2.1 that these primitive probabilities can be written as fractions of atomic probabilities,

$$\tau_i = \frac{\pi_i}{\pi_1 + \pi_2 + \pi_3} \quad \text{and} \quad \tau_i' = \frac{\pi_{\Phi(i)}}{\pi_{\Phi(1)} + \pi_{\Phi(2)} + \pi_{\Phi(3)}} \tag{2.8}$$

for $i = 1, 2, 3$. Now assume that v_1 and v_2 are in the same stage. By definition, the edge labels $\tau_i = \tau_i'$ associated to these vertices are then identified for all $i = 1, 2, 3$. By (2.8), this identification is equivalent to

$$\pi_i(\pi_{\Phi(1)} + \pi_{\Phi(2)} + \pi_{\Phi(3)}) = \pi_{\Phi(i)}(\pi_1 + \pi_2 + \pi_3) \qquad \text{for } i = 1, 2, 3 \tag{2.9}$$

as in (2.6). Cancelling out the terms which appear on both sides of these equations, we obtain that the stage constraints above can be equivalently captured by the following system of polynomial equations

$$\begin{aligned} \pi_1 \cdot (\pi_{\Phi(2)} + \pi_{\Phi(3)}) &= \pi_{\Phi(1)} \cdot (\pi_2 + \pi_3) \\ \pi_2 \cdot (\pi_{\Phi(1)} + \pi_{\Phi(3)}) &= \pi_{\Phi(2)} \cdot (\pi_1 + \pi_3) \\ \pi_3 \cdot (\pi_{\Phi(1)} + \pi_{\Phi(2)}) &= \pi_{\Phi(3)} \cdot (\pi_1 + \pi_2). \end{aligned} \tag{2.10}$$

This is precisely the result of Lemma 2.2 above.

So we have found a system of equations which is equivalent to the graphical constraints apparent in a coloured (staged) probability tree but which does not depend on the parametrisation

of that tree. Whilst these new equations are easy to read off such a graph—simply using the law of conditional probability as above—a given system of this type of polynomial equations is often not very straightforward to interpret. Interestingly, the slight modification we give below enables us to specify an equivalent characterisation of a staged tree model which allows for a very simple probabilistic interpretation.

Definition 2.4 (Odds ratios)**.** In the notation of Lemma 2.2, we specify a collection of equations of the type

$$\pi_i \pi_{\Phi(j)} = \pi_j \pi_{\Phi(i)} \quad \text{for all } i \neq j \tag{2.11}$$

for every two vertices which are in the same stage. Every such identification of products of atomic probabilities is called an *odds-ratio equation.*

Naturally, because in staged trees all atomic probabilities are assumed to be positive, in the definition above the odds-ratio equation (2.11) is equivalent to the identification of the two fractions

$$\frac{\pi_i}{\pi_j} = \frac{\pi_{\Phi(i)}}{\pi_{\Phi(j)}} \quad \text{for all } i \neq j. \tag{2.12}$$

So when these equations are true then π_i relates to π_j as $\pi_{\Phi(i)}$ relates to $\pi_{\Phi(j)}$, for $i \neq j$.

Odds ratios are a frequently used tool in Bayesian inference and gambling (Smith, 2010). Interestingly, from a more methodological point of view, it has also been argued that under certain conditions it can be favourable to elicit odds ratios rather than probability distributions in order to specify a model (Garthwaite et al., 2005). This is not surprising because odds ratios naturally appear when analysing conditional independences in contingency tables (Altham, 1969, 1970a,b), and models determined by this type of conditional independence assumptions are now well studied. When specified as vanishing 2×2 minors of contingency tables, odds ratios can also be interesting objects in Algebraic Statistics (Drton et al., 2009).

The next results paves the way for characterising a staged tree model using precisely systems of odds-ratio equations.

Lemma 2.5 (An equivalent characterisation)**.** *Over the real numbers, equations based on odds ratios (2.11) always imply equations based on the law of conditional probability (2.4). Inside the probability simplex, where $\sum_{i=1}^{n} \pi_i = 1$ for some $n \geq 2k$, the converse holds and (2.4) also imply (2.11). For probability trees, these systems of equations are hence equivalent. Moreover, both are equivalent to $\pi_i = a \cdot \pi_{\Phi(i)}$ for a constant a and all $i = 1, \ldots, k$.*

Proof. '(2.11) imply (2.4)' Assume for all $i \neq j$ that the identification $\pi_i \pi_{\Phi(j)} = \pi_j \pi_{\Phi(i)}$ is true. Then summing over j on both sides of this equation gives us

$$\pi_i \sum_{j \neq i} \pi_{\Phi(j)} = \pi_{\Phi(i)} \sum_{j \neq i} \pi_j \tag{2.13}$$

and adding $\pi_i \pi_{\Phi(i)}$ yields the claim.

'(2.4) imply (2.11)' This is slightly more subtle. Assume (2.4) is true and that atomic probabilities sum to unity, $\sum_{i=1}^{n} \pi_i = 1$. We first denote this sum of all atomic probabilities by

$$\sum_{i=1}^{n} \pi_i = \sum_{t=1}^{k} \pi_t + \sum_{s=1}^{k} \pi_{\Phi(s)} + r = 1 \tag{2.14}$$

where $r = \sum_{j \neq i, \Phi(i) | i=1,\ldots,k} \pi_j$ denotes the remainder term such that the equation above is true. Note that r is constant with respect to the indeterminates in (2.4) and (2.11). Then (2.4) is equivalent to the equation

$$\pi_i \cdot \left(1 - r - \sum_{t=1}^{k} \pi_t\right) = \pi_{\Phi(i)} \sum_{t=1}^{k} \pi_t \tag{2.15}$$

where we replaced one sum of indeterminates by the expression derived in (2.14). Hence, every $\pi_{\Phi(i)}$ can be written entirely in terms of π_i, $i = 1, \ldots, k$, and r:

$$\pi_{\Phi(i)} = \pi_i \cdot \left(\frac{1 - r}{\sum_{t=1}^{k} \pi_t} - 1\right) \qquad \text{for all } i = 1, \ldots, k. \tag{2.16}$$

So the quantity $a = \frac{1-r}{\sum_{t=1}^{k} \pi_t} - 1$ on the right hand side of this expression is the constant we have been looking for: every $\pi_{\Phi(i)}$ is a multiple of every π_i, $i = 1, \ldots, k$.

As a final step, multiplying (2.16) by π_j for some $j \neq i$ and then permuting i and j yields the claim. □

We can see from the proof above that the constant a in Lemma 2.2 is simply a renormalisation constant. Figure 2.3 which illustrates this result. Because the equations (2.11) imply equations (2.4), every point that fulfils (2.11) will also fulfil (2.4). In other words, the solution set[9] $\mathbf{V}_{\text{odds}}$ of (2.11) is contained in the solution set $\mathbf{V}_{\text{poly}}$ of (2.4). After projection onto a space where all indeterminates which are not involved in these equations are constants, one set is a rescaling of the other. Inside the probability simplex, both sets coincide.

9 The choice of the letter $\mathbf{V}$ for labelling this set is intentional: it is indeed a variety. Compare Section 2.2 below.

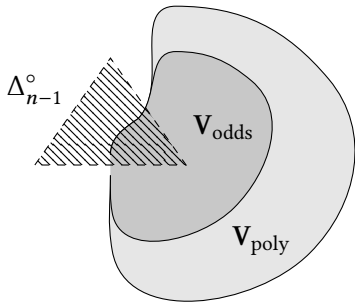


Figure 2.3. An illustration of the solution sets $\mathbf{V}_{\text{odds}}$ of the odds ratios (2.11) and $\mathbf{V}_{\text{poly}}$ of the polynomial equations in (2.4) as characterised in Lemma 2.5.

Of course, none of the solution sets above has to be as smooth as depicted in Fig. 2.3. To our knowledge, the precise geometry of these sets remains an open field of research. Some early results are presented over the next two sections.

Example 2.6 (Example 2.3 continued.)**.** The stage structure of $(\mathcal{T}, \theta_{\mathcal{T}})$ in Fig. 2.2.1 can be characterised by the odds-ratio equations

$$\begin{aligned} \pi_1 \cdot \pi_{\Phi(2)} &= \pi_2 \cdot \pi_{\Phi(1)} \\ \pi_1 \cdot \pi_{\Phi(3)} &= \pi_3 \cdot \pi_{\Phi(1)} \\ \pi_2 \cdot \pi_{\Phi(3)} &= \pi_3 \cdot \pi_{\Phi(2)} \end{aligned} \tag{2.17}$$

as in Lemma 2.5. It can be verified by hand or using any computer algebra programme[10] that when the atomic probabilities sum to unity, $\pi_1 + \pi_2 + \pi_3 + \pi_{\Phi(1)} + \pi_{\Phi(2)} + \pi_{\Phi(3)} = 1$, the above system of equations is indeed equivalent to the system of equations in (2.10). Hereby, the normalisation constant from Lemma 2.5 is given by

$$a = \frac{1}{\pi_1 + \pi_2 + \pi_3} - 1 \tag{2.18}$$

as in (2.16).

The two lemmata above imply our main result in this chapter: any staged tree model can be fully characterised using positivity constraints and sum-to-1 conditions together with a collection of odds-ratio equations. These are exactly the constraints we have been looking for in (2.1).

10 This can be calculated for instance using `CoCoA`: see Section 1.3. We will present some of the necessary commands in the subsequent section.

Theorem 1 (A geometric characterisation of staged tree models)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree. Then the corresponding model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \Delta^{\circ}_{\#\Lambda(\mathcal{T})-1} \cap \{\boldsymbol{p} \in \mathbb{R}^{\#\Lambda(\mathcal{T})} \mid f(\boldsymbol{p}) = 0 \text{ for all } f \in F\}$ equals the intersection of a probability simplex with a system of equations F. These are odds-ratio equations with indeterminates given by probabilities of vertex-centred events, for every two vertices which are in the same stage.*

Proof. This is a direct result of Lemmata 2.2 and 2.5. □

Note that for every two vertices in a staged tree which are in the same stage and have k emanating edges there are k equations arising from the law of conditional probability (2.4) but $\binom{k}{2}$ odds ratios (2.11). Because these are equivalent, k is a lower bound on the number of equations we need to specify in order to capture a single stage constraint. It is easy to check that for instance in (2.10) the latter two equations imply the first. It follows that some of these k equations might still be redundant so our characterisation is not necessarily minimal. We will see in the subsequent section how tools from computational algebra can be used to specify a minimal system of equations in the characterisation given by the theorem above.

In the example in the introduction from page 3 we have seen that a staged tree model on three atoms representing a repeated coin toss could be drawn as a parametric curve inside a probability simplex. Theorem 1 now yields an equivalent characterisation of that curve as the solution set of a collection of odds-ratio equations between atomic probabilities.

Example 2.7 (A repeated coin toss repeated)**.** Consider again the binary staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 0.1.1. Here, the root v_0 and the inner vertex v_1 are in the same stage. Then Theorem 1 yields that the associated model

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \left\{\left(\theta^2, \theta(1-\theta), 1-\theta\right) \mid \theta \in (0,1)\right\} \tag{2.19}$$

as specified in (0.1) equals the intersection

$$\Delta^{\circ}_{3-1} \cap \left\{(p_1, p_2, p_3) \in \mathbb{R}^3 \mid p_1 p_3 = p_2(p_1 + p_2)\right\} \tag{2.20}$$

of the two-dimensional probability simplex in $\mathbb{R}^3$ with the solution set of the polynomial equation $f(\boldsymbol{p}) = p_1 p_3 - p_2(p_1 + p_2) = 0$ for $\boldsymbol{p} = (p_1, p_2, p_3) \in \mathbb{R}^3$. This intersection is precisely the curve depicted in Fig. 0.1.2.

So for all points $\boldsymbol{p}$ in the model:

$$\frac{p_1}{p_1 + p_2} = \frac{p_2}{p_3}. \tag{2.21}$$

This constraint simply states that the odds of seeing a double 'head' against seeing 'head' at all are the same as seeing 'head, tails' against 'tails'. This is therefore precisely equivalent to

the independence constraint we introduced in order to define this model: the second coin toss is independent of the first.

By construction, the new characterisation of staged tree models provided in Theorem 1 is independent of a graphical representation in terms of a given staged tree[11]. We will provide a brief illustration of the implications of this result here and defer a full analysis to the next chapter.

Example 2.8 (Example 2.3 continued.)**.** The staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ in Fig. 2.2.2 represents the same statistical model as the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 2.2.1. This can be easily seen by verifying that both staged trees share the same tree parametrisation—we have illustrated this in translating the colouring of vertices across both figures. In fact, this model is given by

$$\begin{aligned}\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \Big\{\big(\theta_0\tau_0, \theta_0\tau_1, \theta_0\tau_2, \theta_1\tau_0, \theta_1\tau_1, \theta_1\tau_2\big) \mid \tau_0 + \tau_1 + \tau_2 = 1,\ \theta_0 + \theta_1 = 1 \text{ and} \\ \tau_i, \theta_j \in (0,1), i = 0,1,2; j = 0,1\Big\}.\end{aligned} \tag{2.22}$$

By Lemma 2.2, the stage structure of $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ can be captured using the odds-ratio equations

$$\begin{aligned}\pi_1 \cdot \pi_{\Phi(2)} &= \pi_2 \cdot \pi_{\Phi(1)} \\ \pi_1 \cdot \pi_{\Phi(3)} &= \pi_3 \cdot \pi_{\Phi(1)} \\ \pi_2 \cdot \pi_{\Phi(3)} &= \pi_3 \cdot \pi_{\Phi(2)}.\end{aligned} \tag{2.23}$$

By Lemma 2.5, inside the probability simplex these equations are precisely equivalent to the characterising equations in (2.10) obtained for $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. As a consequence, the staged tree model at hand can be characterised using two different but equivalent systems of equations,

$$\begin{aligned}\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} &= \{\boldsymbol{p} \in \mathbb{R}^6 \mid f(\boldsymbol{p}) = 0 \text{ for all polynomials } f \text{ as in (2.10)}\} \cap \Delta^{\circ}_{6-1} \\ &= \{\boldsymbol{p} \in \mathbb{R}^6 \mid g(\boldsymbol{p}) = 0 \text{ for all polynomials } g \text{ as in (2.23)}\} \cap \Delta^{\circ}_{6-1} = \mathbb{P}_{(\mathcal{S},\boldsymbol{\theta}_{\mathcal{S}})}\end{aligned} \tag{2.24}$$

arising from two different graphical representations $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ of the same model.

To conclude this section, note that we can easily use the characterisation obtained above in order to prove a statement we have made in Section 1.3.2 above: that when introducing additional stages into a staged tree, the new graph represents a submodel of the one represented by the original tree. This is well known and intuitively clear: whenever a set of probability

11 This characterisation is thus the same for all statistically equivalent staged trees (Definition 3.10) and is invariant to an application of the swap (Definition 3.22) and resize operator (Definition 3.33) on a given graphical representation.

distributions fulfils all the constraints imposed by a model A and possibly also a collection of additional constraints B, then the distributions fulfilling both A and B form a submodel of the model A. However, in staged tree models this result is not very straightforward to prove using a parametric approach. In particular, working purely on Definition 1.7 we would need to show that a projection on the parameter space of a tree parametrisation translates into a subset constraint on the image of that map. We can now show this result much more straightforwardly as presented below.

Corollary 2.9 (Submodels of staged tree models)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree. Denote by $(\mathcal{T}', \boldsymbol{\theta}_{\mathcal{T}'})$ a staged tree with the same graph $\mathcal{T} = \mathcal{T}'$ but additional stage structure, such that every two vertices which are in the same stage in $\mathcal{T}$ are also in the same stage in $\mathcal{T}'$. Then the new model $\mathbb{P}_{(\mathcal{T}', \boldsymbol{\theta}_{\mathcal{T}'})} \subseteq \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$ is a submodel of the original.*

Proof. Every stage constraint in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ specifies a collection of polynomials $F_{\mathcal{T}}$ as in Theorem 1. Within the probability simplex, the tree model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$ is characterised as the solution set of these polynomials set to zero, so $\{\boldsymbol{p} \mid f(\boldsymbol{p}) = 0 \text{ for all } f \in F_{\mathcal{T}}\}$. Now by construction, $(\mathcal{T}', \boldsymbol{\theta}_{\mathcal{T}'})$ induces these same equations plus an additional system of equations as in Lemma 2.2 for every two vertices which are in the same stage in $\mathcal{T}'$ but not in $\mathcal{T}$. So $\mathbb{P}_{(\mathcal{T}', \boldsymbol{\theta}_{\mathcal{T}'})}$ is the zero set of polynomials in a collection $F_{\mathcal{T}'} \supseteq F_{\mathcal{T}}$ which include the originally specified polynomials. Because every point which is a zero of all equations induced by $F_{\mathcal{T}'}$ is clearly also a zero of those induced by $F_{\mathcal{T}}$, we obtain that one model is a subset of the other, $\mathbb{P}_{(\mathcal{T}', \boldsymbol{\theta}_{\mathcal{T}'})} \subseteq \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$. The claim follows. □

The above result will be employed in Section 2.3 where we analyse the relationships between various staged tree models on the same number of atoms.

2.2. Staged trees as algebraic statistical models

We will now use the vocabulary introduced in Section 1.3 and present methods from algebraic geometry which can be of great help in the specification of staged tree models. We can then translate the results obtained above into a language that enables us to compute the equations in the characterisation in Theorem 1 from a given parametrised vector of atomic probabilities.

Following Drton and Sullivant (2007), an *algebraic statistical model* is a parametric statistical model which can be characterised using semi-algebraic sets: so a model which is specified as the solution set of polynomial equations and inequalities as in (1.29). Surprisingly many models are algebraic statistical models, including many conditional independence models and

all models whose parameter spaces are constrained by rational functions. In these models statistical inference can often be performed using tools from algebra and algebraic geometry. These tools have already been employed over a wide range of successful applications, a number of which we have cited in Section 1.3.2 above. For convenience, we repeat here the for us most important[12] result, namely that decomposable Bayesian networks are algebraic statistical models:

Proposition 2.10 (Theorem 4.3 of Geiger et al. (2006))**.** *Let G be a decomposable graphical model. Then the set of quadratic binomials representing cross-product differences for saturated conditional independence statements for G forms a Gröbner basis of the toric ideal $I_{A(G)}$.*

Translated into the somewhat more superficial language we use in this thesis, the result above reads: *Every conditional independence statement characterising a decomposable model can be equivalently captured using a collection of cross-product differences. These define a very special set of generators of an ideal whose induced variety contains the distributions in the model, and this ideal is toric.* Toric equations are of the type (1.30). A Gröbner basis is a generating set of an ideal with a number of very desirable properties: see Chapter 2 in Cox et al. (2015). An in-depth analysis of the algebraic properties of the toric ideal above and of its induced variety, including a full specification of the algebraic properties of all conditional independence models on five or less random variables, has been provided by Garcia et al. (2005). The importance of toric ideals and varieties in statistical inference has also been highlighted by Pistone et al. (2006).

Proposition 2.10 is one of many entries in the dictionary between algebra and statistics that has been set up by the community of researchers in Algebraic Statistics over the past two decades. We can observe here a theme which resonates throughout this field: that for many objects which are of interest to statistics, there is an equally interesting object in algebraic geometry. In fact, many notions in these two fields translate seamlessly, and it often turns out that the steps taken in a statistical analysis of an object are exact duals of the steps taken in an algebraic analysis of 'the same' object. For instance, a Bayesian network model can be specified either using a recursive factorisation of a probability distribution or a collection of conditional independence constraints. And in the same way, an algebraic variety can be specified either parametrically or implicitly (see below). These notions are duals of each other because by the result above Bayesian networks 'are' toric varieties. In the exact same way, we have seen above that staged tree models can be characterised using either a tree parametrisation or the solution sets of odds-ratio equations. So in fact, Theorem 1 is the direct analogue to Proposition 2.10, and

12 The paper by Geiger et al. (2006) characterises a large number of well-known models as algebraic-statistical models, even though the terminology used by these authors is different. We discuss some of their results, including an interpretation of staged tree models as log-linear models, in Görgen et al. (2017).

provides a generalisation of this link between algebra and statistics from Bayesian networks to staged tree models.

Now, although we have implicitly already obtained that result in the previous section, we will first show below that staged tree models are algebraic statistical models. We then introduce the tools from algebraic geometry which can be employed in the study of these models, and we finally validate our results using the proposition above.

By Definition 1.5, the parameter space of a probability tree is the Cartesian product of semi-algebraic sets of the form

$$\left\{(p_1,\dots,p_n)\in\mathbb{R}^n \mid \sum_{i=1}^{n} p_i - 1 = 0 \text{ and } p_i > 0,\ p_i - 1 < 0 \text{ for all } i = 1,\dots,n\right\} \tag{2.25}$$

for some $n \in \mathbb{N}$ as in (1.1). The celebrated *Tarski-Seidenberg Theorem* which is of huge importance to Algebraic Statistics[13] states that the image of a semi-algebraic set under a rational function is again semi-algebraic (Benedetti and Risler, 1990). Hence, because the parametrisation of probability tree models is of monomial form—and so is in particular a rational function—all probability tree models are algebraic statistical models. The interesting question however is of what type the constraining equations and inequalities are, and what the nature of these equations can tell us about the modelling assumptions.

We have shown in Theorem 1 that the equations describing staged tree models are based on odds ratios and the inequalities are given by positivity constraints. In order to obtain this result, we have performed a number of calculations which assumed that we can divide by a positive number and cancel factors in the numerator and denominator of a fraction. However, if we want to use computational algebra software in order to derive these constraints from a given tree parametrisation, we can not offhandedly assume that these notions are readily transferable. On the contrary, much care is needed when employing the tools provided by algebraic geometry in order to translate a monomial parametrisation of a discrete statistical model into a collection of polynomial constraints on points in a real space. We provide details of this procedure in this section, following the theory presented in Chapter 3 of Cox et al. (2015).

Let $\mathbb{P}_{(\mathcal{T},\theta_{\mathcal{T}})}$ denote a staged tree model with tree parametrisation $\Psi_{\mathcal{T}} : \bigtimes_{j=1}^{m} \Delta^{\circ}_{n_j-1} \to \Delta^{\circ}_{n-1}$ in the notation of Definitions 1.2 and 1.7. The components in the image of that map are given by the monomials $\Psi_{\mathcal{T},i}(\boldsymbol{\theta}) = \boldsymbol{\theta}^{\alpha_i}$ or atomic probabilities of n atoms, so $i = 1,\dots,n$. Then let

$$g_i(\boldsymbol{p},\boldsymbol{\theta}) = p_i - \boldsymbol{\theta}^{\alpha_i} \qquad \text{for } i = 1,\dots,n \tag{2.26}$$

13 See for instance Drton and Sullivant (2007) and references therein.

be a collection of polynomials given by the differences between one component of a generic point $\boldsymbol{p} = (p_1, \ldots, p_n) \in \mathbb{R}^n$ and an atomic monomial in the image of the tree parametrisation above. Simply by construction, a distribution $\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}}$ now lies in the tree model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$ if and only if it there exists a choice of the parameters $\boldsymbol{\theta}^{\alpha_i} \in \bigtimes_{j=1}^{m} \Delta^\circ_{n_j-1}$ in the domain of $\Psi_\mathcal{T}$ such that the equations $g_i(\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}}, \boldsymbol{\theta}) = 0$ vanish for all $i = 1, \ldots, n$. As a consequence, when cancelling the $\boldsymbol{\theta}$-indeterminates from this almost tautological characterisation of the model, we obtain a new system of equations which characterises the points $\boldsymbol{p} = \boldsymbol{\pi}$ in this model independent of a given parametrisation.

The process of cancelling a choice of indeterminates out of a collection of equations is known as the method of *implicitisation* in algebraic geometry. This method essentially performs a projection from a polynomial ring in the indeterminates $\boldsymbol{p}$ and $\boldsymbol{\theta}$ onto a ring in the indeterminates $\boldsymbol{p}$, without $\boldsymbol{\theta}$. Over a field—rather than over a probability simplex—calculations of this type can be performed with any computer algebra software. We give examples of this below. Hereby, an *explicit* characterisation or parametrisation of a set of points $p_i = \boldsymbol{\theta}^{\alpha_i}$, $i = 1 \ldots, n$, is transformed into an equivalent *implicit* system $f_j(\boldsymbol{p}) = 0$, $j = 1, \ldots, k$. The main advantage of an explicit system of equations is that it enables us to straightforwardly draw the corresponding model as the image of a parametrisation map: we simply substitute values for $\boldsymbol{\theta}$ and obtain points $\boldsymbol{p}$ in the image. An implicit description on the other hand has the advantage that we can easily check whether a point $\boldsymbol{p}$ lies in the model: simply by verifying that all functions $f_j(\boldsymbol{p}) = 0$ vanish at that point, $j = 1, \ldots, k$. Without implicitisation, in order to determine whether a point lies on the model we would need to find an inverse map to a given parametrisation—this is often much harder. As a consequence, implicitisation methods and the result of Theorem 1 can be used in questions of model fit. Drton and Sullivant (2007) first presented this idea for Gaussian models.

So when developing the result of Theorem 1 using the language of algebraic geometry, we aim to answer the question: *Which equations are obtained when eliminating $\boldsymbol{\theta}$ from $g_i(\boldsymbol{p}, \boldsymbol{\theta}) = 0$ for all $i = 1, \ldots, n$?*

We have of course already seen what the answer to this question *should be*: the equations F are given by (2.1), so are based on odds ratios. However, the technical subtleties mentioned in Section 1.3 which are involved in an algebraic approach to this problem might yield a different characterisation. This is mainly due to the above stated fact that implicitisation needs to be performed over a field. So in order to be able to employ computer algebra software, the domain of a tree parametrisation would need to be assumed to be $\mathbb{R}^d$ or at least $\mathbb{Q}^d$ rather than a product of open probability simplices $\bigtimes_{j=1}^{m} \Delta^\circ_{n_j-1}$ as above. In the algebraic characterisations of statistical models developed by other authors, positivity constraints are often dealt with by

working on 'closed' sets instead. These are sets which do not allow for inequalities: we develop on this point below.

Our first result calls on the tools needed for transforming the equations specified above.

Lemma 2.11 (Implicitisation). *Let $\Psi_{\mathcal{T}} : \mathbb{R}^d \to \mathbb{R}^n, \boldsymbol{\theta} \mapsto \boldsymbol{\pi}$ denote the parametrisation map of a labelled event tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with real-valued labels $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathcal{T}}$. Let then*

$$I_{\Psi_{\mathcal{T}}} = \langle \pi_i - \Psi_{\mathcal{T},i}(\boldsymbol{\theta}) \mid i = 1, \ldots, n \rangle \subseteq \mathbb{R}_{d+n}[\boldsymbol{\theta}, \boldsymbol{\pi}] \tag{2.27}$$

denote the ideal generated by the equations which determine points in the image of the tree parametrisation. Then the variety $\mathbf{V}_{\Psi_{\mathcal{T}}} = \mathbf{V}(I_{\Psi_{\mathcal{T}}})$ where these equations vanish is the graph of the function $\Psi_{\mathcal{T}}$.

The elimination ideal *$I_{\mathcal{T}} = I_{\Psi_{\mathcal{T}}} \cap \mathbb{R}_n[\boldsymbol{\pi}]$ induces a variety $\mathbf{V}_{\mathcal{T}} = \mathbf{V}(I_{\mathcal{T}})$ which is the smallest variety in $\mathbb{R}^n$ containing the image of the parametrisation $\Psi_{\mathcal{T}}$.*

Proof. Theorem 1 (Polynomial Implicitisation) in Section 3.3 of Cox et al. (2015). □

The result of Lemma 2.11 is twofold. First, it provides us with an algebraic counterpart for an insight that is immediate in parametric statistical models: that we can actually draw these models as graphs of certain functions, namely their parametrisations. Second, the lemma hands us the algebraic tools to determine points on this graph and so possible elements of the corresponding parametric statistical model. The resulting strategy is as follows. When given a parametrised tree model we specify the ideal (2.27) of differences which determine points in the model using equations (1.14) and (2.26). We then intersect this ideal with a lower-dimensional polynomial ring which does not depend on the parametrisation: we *eliminate* the parameters. The resulting set of equations is toric because our parametrisation was a monomial map. As a final step, we determine the variety of that ideal, so the set of common zeros of the generating polynomials. The variety we obtain in this way is by construction the smallest variety containing the image of the given parametrisation map. The notion of a 'small' variety hereby refers to smallness with respect to set inclusion: any other variety containing that image contains also the variety we computed. The variety of the elimination ideal is in that sense the closest (algebraic) approximation to a statistical model with the given parametrisation[14].

The technical challenge in implementing the procedure outlined above lies in the projection step that determines the elimination ideal $I_{\Psi_{\mathcal{T}}} \cap \mathbb{R}_n[\boldsymbol{\pi}]$ which effects the cancelling operation of

14 This approximation is in fact the *Zariski closure* of the image of the parametrisation: see also Theorem 3 (Closure Theorem) in Section 3.2 of Cox et al. (2015). The Zariski closure of any set $S \subseteq \mathbb{R}^d$ can be defined as the variety of the set of polynomials which vanish at the points in that set, so $\mathbf{V}(I(S))$. So the closure of a set defined by polynomial inequalities is a set determined by polynomial equations. For instance, the Zariski closure of the ray $\mathbb{R}_{\geq 0}$ is all of $\mathbb{R}$, and the Zariski closure of a segment of a graph is the entire graph.

the indeterminates given by the parametrisation. In particular, here we would need to specify a special set of polynomials—in fact a Gröbner basis—generating the ideal $I_{\Psi_\mathcal{T}}$ and then project these generators onto the lower dimensional ring. Section 3.3 of Cox et al. (2015) presents an algorithm for performing this type of projection and an efficient implementation for toric ideals has been presented in Section 12 of Sturmfels (1996). These have now been implemented in a large range of software.

Importantly, because of the simplified assumptions we have made above, the variety we specify using implicitisation often does not equal the model itself. It is in general very difficult to determine the set difference between that variety and the actual image of a tree parametrisation. Examples and solutions in very special cases are given in Chapter 3 of Cox et al. (2015). So in the following we will often find that a staged tree model we wish to characterise

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})} = \mathbf{V}(I_{\text{odds}}) \cap \Delta^\circ_{n-1} \subseteq \mathbf{V}(I_\mathcal{T}) \cap \Delta^\circ_{n-1} \tag{2.28}$$

is a proper subset of the algebraic characterisation we find. Equivalently, computations using computer algebra as performed below might yield an ideal $I_\mathcal{T} \subseteq I_{\text{odds}}$ which is smaller than the one desired, where $\mathbf{V}(I_{\text{odds}}) = \mathbf{V}_{\text{odds}}$ denotes the solution set from Theorem 1 and Fig. 2.3. We will henceforth call the elimination ideal $I_\mathcal{T}$ obtained after implicitisation in Lemma 2.11 a *model ideal*. This is in close analogy to the terminology used in Drton and Sullivant (2007) where ideals characterising a model were named *ideals of model invariants*[15].

As a direct result of Theorem 1, we obtain that staged tree models can be viewed as intersections of toric varieties with affine hyperplanes:

Corollary 2.12 (The model ideal)**.** *Let $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})$ be a staged tree. Then the model ideal $I_\mathcal{T}$ together with the sum-to-*1 *conditions is contained in an ideal which is generated by equations of the form*

$$\langle \Big(\sum_{i\in I}\pi_i\Big)\Big(\sum_{l\in L}\pi_l\Big) - \Big(\sum_{k\in K}\pi_k\Big)\Big(\sum_{j\in J}\pi_j\Big) \mid I,J,K,L\rangle + \langle \sum_{t=1}^{n}\pi_t - 1\rangle \tag{2.29}$$

where $I,J,K,L \subseteq \{1,2,\ldots,n\}$ are index sets arising from stage constraints on $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})$, as in Lemma 2.2.

Interestingly, the image of every rational parametrisation has an implicit description as in Lemma 2.11 and Corollary 2.12 but conversely a parametrisation can be found only for very specific solution sets of systems of polynomial equations: these are called *unirational* varieties.

15 This terminology in turn arised from an application in phylogenetics where these ideals were used to determine so-called *phylogenetic invariants*: see also Pachter and Sturmfels (2005).

By the above, a staged tree is a vehicle which enables us to both read off an explicit parametrisation and to read off implicit constraints in terms of odds ratios. We can thus think of staged trees as illustrations of an inverse map between these two alternative ways of describing the same system. An even stronger result is that whenever there is a staged tree description of a parametrisation then the corresponding variety is unirational. In the Chapter 3 below we will prove conditions under which a given parametrisation is a tree parametrisation.

We will dedicate the remainder of this section to an analysis of two model ideals obtained after implicitisation in staged tree models. The first example is a small scale illustration of the repeated coin-toss model we have seen before, and the second example embeds our results in the current literature. The programme `CoCoA` will be used to compute intersections of ideals and polynomial rings as above and to obtain the result of Lemma 2.11 in the example settings. Hereby, the command `elim(t[0]..t[d],I)` eliminates all parameters `t[0]`,...,`t[d]` from an ideal `I` as in (2.27).

Example 2.13 (Example 2.7 continued)**.** The parametrisation of the repeated coin-toss model represented by the tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 0.1 is given by the map

$$\Psi_{\mathcal{T}} : (\theta_0, \theta_1) \mapsto (\theta_0^2, \theta_0\theta_1, \theta_1). \tag{2.30}$$

In contrast to the original specification in (0.2), we now assume that the domain of that map is $\mathbb{R}^2$ rather than Δ°_{2-1}. Then the ideal arising from the equations $\Psi_{\mathcal{T},i} = \pi_i$, $i = 1, 2, 3$, which specify points in this model is given by

$$I_{\Psi_{\mathcal{T}}} = \langle \pi_1 - \theta_0^2, \pi_2 - \theta_0\theta_1, \pi_3 - \theta_1 \rangle \tag{2.31}$$

as a subset of the polynomial ring $\mathbb{R}[\theta_0, \theta_1, \pi_1, \pi_2, \pi_3]$ in all indeterminates in the characterisation above. The variety

$$\mathbf{V}(I_{\Psi_{\mathcal{T}}}) = \{(\boldsymbol{\theta}, \boldsymbol{\pi}) \in \mathbb{R}^5 \mid f(\boldsymbol{\theta}, \boldsymbol{\pi}) = 0 \text{ for all } f \in I_{\Psi_{\mathcal{T}}}\} \tag{2.32}$$

is then the set of all points $(\theta_0, \theta_1, \pi_1, \pi_2, \pi_3)$ for which the equations above vanish. In fact, by Lemma 2.11, this is simply the graph of $\Psi_{\mathcal{T}}$. We draw a detail of this graph in Fig. 2.4. Notably, the image of $\Psi_{\mathcal{T}}$ is now a two-dimensional surface in three dimensions and is thus very different from the one-dimensional curve we have drawn in Fig. 0.1.2 where the parametrisation's domain was restricted to the probability simplex.

We can now use the following `CoCoA` code to eliminate θ_0 and θ_1 from the ideal in (2.31):

```
1 Use R ::= QQ[t[0..1],p[1..3]];
```

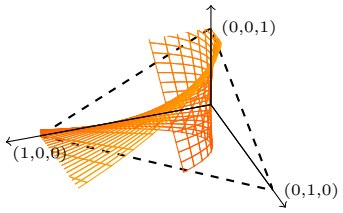


Figure 2.4. The image of the parametrisation in (2.30) is not contained in the probability simplex if constraints on its domain are relaxed. We depict this here for $(\theta_0, \theta_1) \in [-0.5, 1.5]^2$.

```
Ipsi := Ideal(p[1]-t[0]*t[0],p[2]-t[0]*t[1],p[3]-t[1]);
IT := elim(t[0]..t[1],I);
```

This elimination step yields as a result the model ideal

$$I_{\mathcal{T}} = I_{\Psi_{\mathcal{T}}} \cap \mathbb{R}[\pi_1, \pi_2, \pi_3] = \langle \pi_2^2 - \pi_1(\pi_3 - \pi_1) \rangle \tag{2.33}$$

in the polynomial ring $\mathbb{R}[\pi_1, \pi_2, \pi_3]$ whose indeterminates are atomic probabilities. Note that as desired the model ideal $I_{\mathcal{T}}$ does not depend on θ_0 or θ_1 but is characterised using only components of points in the image of the parametrisation as indeterminates.

Comparing the above result with our previous analysis of this example, we observe that the model ideal (2.33) is not generated by the polynomial $\pi_1\pi_3 = \pi_2(\pi_1 + \pi_2)$ we found in Example 2.7. So the elimination procedure has not been able to identify the correct odds-ratio equations. However, by Corollary 2.12 above, after imposing sum-to-1 conditions the model ideal is still contained in the ideal specified by the odds ratios of Theorem 1, so

$$I_{\mathcal{T}} + \langle \pi_1 + \pi_2 + \pi_3 - 1 \rangle \subseteq I_{\text{odds}} + \langle \pi_1 + \pi_2 + \pi_3 - 1 \rangle. \tag{2.34}$$

where again $I_{\text{odds}} = \langle \pi_1\pi_3 - \pi_2(\pi_1 + \pi_2) \rangle$ is generated by the polynomial from (2.20).

We can check that this is the case simply by extending the CoCoA code provided above to

```
sumto1 := p[1]+p[2]+p[3]-1;
S := Ideal(sumto1);
Iodds := Ideal(p[1]*p[3]-p[2]*(p[1]+p[2]));
IsContained(IT+S,Iodds+S);
```

and checking that the final line returns the value `true`.

The line

```
IsContained(Iodds+S,IT+S);
```

which returns the value `false` confirms that the inclusion above is strict: the model ideal is not equal to the ideal specified in terms of odds ratios.

As a consequence, we have thus shown that the staged tree model we wish to specify is contained in the variety induced by the model ideal,

$$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})} = \mathbf{V}(\pi_1\pi_3 - \pi_2(\pi_1+\pi_2)) \cap \Delta^\circ_{3-1} \subseteq \mathbf{V}(\pi_2^2 - \pi_1(\pi_3-\pi_1)) \cap \Delta^\circ_{3-1}. \tag{2.35}$$

This result is by Lemma 2.11 the best algebraic approximation to the model we aim to characterise.

So can we do better than this? Interestingly, if we impose the *local* sum-to-1 condition $\theta_0 + \theta_1 = 1$ on the tree before implicitisation, then the set inclusion above becomes an equality. The code

```
floretsumto1 := t[0]+t[1]-1;
F := Ideal(floretsumto1);
newIT := elim(t[0]..t[1],I+F);
newIT=Iodds+S;
```

which returns `true` for the equality of these ideals confirms that after implicitisation of the new ideal $I_{\Psi_\mathcal{T}} + \langle\theta_0 + \theta_1 - 1\rangle$ we obtain that

$$\langle \pi_1 - \theta_1^2, \pi_2 - \theta_1\theta_2, \pi_3 - \theta_2, \theta_1 + \theta_2 - 1\rangle \cap \mathbb{R}[\pi_1,\pi_2,\pi_3] = I_{\text{odds}} + \langle \pi_1+\pi_2+\pi_3-1\rangle. \tag{2.36}$$

So this new implicitisation yields precisely odds-ratio differences and global sum-to-1 conditions on atomic probabilities. Hence, when taking a priori sum-to-1 conditions into account, we can obtain the correct staged tree model (up to positivity constraints).

Sadly, the ansatz of including floret sum-to-1 conditions before implicitisation does not provide a general solution to the problem of finding an exact algebraic description of a staged tree model: the models analysed in the subsequent section provide numerous examples where floret sum-to-1 conditions are *not* sufficient for improving our algebraic approximation of a staged tree model. So the ideal inclusions and set inclusions in (2.28) above are strict in these cases. The problem of completely specifying these models using only elimination theory and local or global sum-to-1 conditions provides a very rich vein of research and remains open at the time of writing.

We have shown in Section 1.2 that every discrete Bayesian network can be represented by a staged tree instead and that the class of all BN models is a subclass of the class of all staged tree models. Now by Proposition 2.10, discrete and decomposable Bayesian networks are character-

ised by toric polynomial equations. So for all staged tree models which are also decomposable BN models, our results should be equivalent to the one stated above.

So as a final result in this section we will provide a short illustration showing that our results are in line with the results found by the authors we have cited. Interestingly, while their toric characterisation relied on a proof which went through all single steps in the polynomial implicitisation and effectively constructed the polynomials generating the corresponding elimination ideal (Hoşten and Sullivant, 2002), our odds-ratio equations could very straightforwardly be read directly from the graph.

Example 2.14 (Example 1.3 continued)**.** For $i, j, k = 0, 1$ let

$$g_{ijk}(\boldsymbol{p}, \boldsymbol{\theta}) = p_{ijk} - \theta_{ij}\theta_{ijk} \tag{2.37}$$

be a collection of polynomials which arise from the parametrisation of a conditional independence model on three binary random variables as given in (1.9). Then a point $\boldsymbol{p} \in \Delta^\circ_{8-1}$ in the probability simplex lies in this model if and only if there exists a vector $\boldsymbol{\theta} \in \Delta^\circ_{4-1} \times \Delta^\circ_{8-1}$ of parameters such that the equations $g_{ijk}(\boldsymbol{p}, \boldsymbol{\theta}) = 0$ vanish for all $i, j, k = 0, 1$.

Using `CoCoA` code analogous to the one provided in the example above, we can again calculate the elimination ideal $\langle p_{ijk} - \theta_{ij}\theta_{ijk} \mid i, j, k = 0, 1\rangle \cap \mathbb{R}[p_{ijk} \mid i, j, k = 0, 1]$ of the ideal generated by the differences in (2.37). This ideal is a subset of the polynomial ring which does not depend on the given parametrisation. The result of this implicitisation step is the model ideal

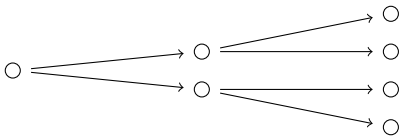
$$\langle p_{000}p_{101} - p_{001}p_{100},\ p_{010}p_{111} - p_{011}p_{110}\rangle. \tag{2.38}$$

So this model ideal is precisely the ideal generated by the cross-product differences found in (1.6) and (1.7). As a consequence, our approach from Lemma 2.11 in this example yields the odds ratios from Theorem 1 and these are toric equations of degree 2, or equations which determine points on the well-known *Segre* variety (Drton et al., 2009).

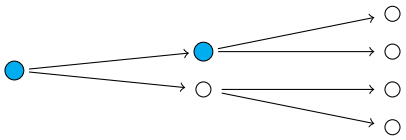
2.3. All staged tree models on four atoms

We will now provide a brief analysis of all staged tree models on four atoms. These models are sufficiently small in dimension to enable us to draw them as sets of points inside the three-dimensional probability simplex Δ°_{4-1} in $\mathbb{R}^3$.

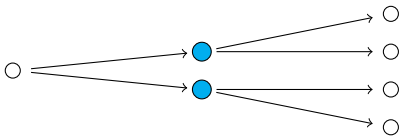
In order to determine these models, consider first Fig. 2.5 which shows all staged trees with precisely four root-to-leaf paths. We omitted here multiple depictions of saturated probability trees and of staged trees which are equal up to a renaming of their edge labels. So for simplicity,



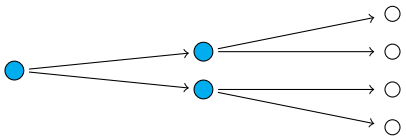
(2.5.1) A saturated probability tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_1$.



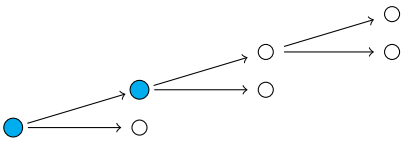
(2.5.2) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_2$.



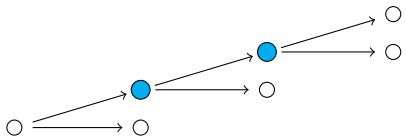
(2.5.3) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_3$ representing a binary independence model.



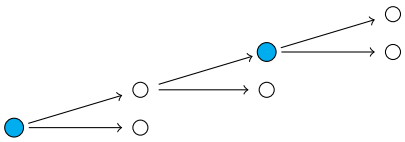
(2.5.4) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_4$.



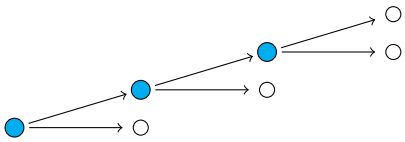
(2.5.5) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_5$.



(2.5.6) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_6$.



(2.5.7) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_7$.



(2.5.8) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_8$.

Figure 2.5. All staged trees with four root-to-leaf paths. The blue circles indicate vertices in the same stage.

we have dropped all labels on these graphs but only indicated the relevant stages in blue colour. Then the analysis below characterises the models represented by these staged trees up to a numbering of the atoms, so up to a rotation of these models inside the probability simplex.

In the following we always denote by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_i$ the staged tree depicted in Fig. 2.5.i and denote the model represented by that tree by $\mathbb{P}_i = \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_i}$ for all $i = 1, \ldots, 8$.

Observe first that a staged tree with four root-to-leaf paths has at most three levels, so root-to-leaf paths with at most three edges. The saturated tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_1$ and the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_3$ representing an independence model[16] are the only square-free staged trees in this class. All other staged trees with four root-to-leaf paths have at least two vertices which are in the same stage and which are also connected by a root-to-leaf path. Explicitly, the graphs in Fig. 2.5 then

16 In fact, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_3$ is (X_1, X_2)-compatible with two binary and independent random variables $X_1 \perp\!\!\!\perp X_2$.

represent the following parametric models:

$$\mathbb{P}_1 = \left\{\left(\theta_0\tau_0, \theta_0\tau_1, \theta_1\sigma_0, \theta_1\sigma_1\right) \mid (\theta_0,\theta_1),(\tau_0,\tau_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.1}$$

$$\mathbb{P}_2 = \left\{\left(\theta_0^2, \theta_0\theta_1, \theta_1\sigma_0, \theta_1\sigma_1\right) \mid (\theta_0,\theta_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.2}$$

$$\mathbb{P}_3 = \left\{\left(\theta_0\sigma_0, \theta_0\sigma_1, \theta_1\sigma_0, \theta_1\sigma_1\right) \mid (\theta_0,\theta_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.3}$$

$$\mathbb{P}_4 = \left\{\left(\theta_0^2, \theta_0\theta_1, \theta_1\theta_0, \theta_1^2\right) \mid (\theta_0,\theta_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.4}$$

$$\mathbb{P}_5 = \left\{\left(\theta_0^2\sigma_0, \theta_0^2\sigma_1, \theta_0\theta_1, \theta_1\right) \mid (\theta_0,\theta_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.5}$$

$$\mathbb{P}_6 = \left\{\left(\theta_0\sigma_0^2, \theta_0\sigma_0\sigma_1, \theta_0\sigma_1, \sigma_1\right) \mid (\theta_0,\theta_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.6}$$

$$\mathbb{P}_7 = \left\{\left(\theta_0^2\sigma_0, \theta_0\sigma_0\theta_1, \theta_0\sigma_1, \theta_1\right) \mid (\theta_0,\theta_1),(\sigma_0,\sigma_1) \in \Delta^\circ_{2-1}\right\} \tag{2.39.7}$$

$$\mathbb{P}_8 = \left\{\left(\theta_0^3, \theta_0^2\theta_1, \theta_0\theta_1, \theta_1\right) \mid (\theta_0,\theta_1) \in \Delta^\circ_{2-1}\right\}. \tag{2.39.8}$$

Because $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_1$ is a saturated tree, $\mathbb{P}_1 = \Delta^\circ_{4-1}$ above is the saturated model on four atoms and is equal to the entire probability simplex[17]. The other models are subsets of that simplex. We will now investigate how $\mathbb{P}_2,\ldots,\mathbb{P}_8$ can be specified implicitly as solution sets of odds-ratio equations as in Theorem 1. This characterisation—based on the results we developed in Sections 2.1 and 2.2 above—will then enable us to analyse whether the models above are pairwise different and how they relate to each other. Note that these two queries cannot be straightforwardly answered using the parametrisations given in (2.39).

We will first provide some graphical intuition and then prove our results using algebra below. Note that $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_2$, $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_3$ and $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_4$ in Figs. 2.5.2 to 2.5.4 share the same event tree $\mathcal{T}_2 = \mathcal{T}_3 = \mathcal{T}_4$ with different stage structure. In particular, both the stages of $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_2$ and of $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_3$ are also present in $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_4$. So by Corollary 2.9, the corresponding model $\mathbb{P}_4$ must be a submodel of both $\mathbb{P}_2$ and $\mathbb{P}_3$. But how exactly does $\mathbb{P}_4$ relate to $\mathbb{P}_2$ and $\mathbb{P}_3$? With a little intuition on staged trees, we can guess[18] that $\mathbb{P}_4$ might correspond to an intersection of the other two models $\mathbb{P}_2 \cap \mathbb{P}_3$ together with a rotation of $\mathbb{P}_2$ inside the simplex. This is because when we 'overlay' the colourings of the staged trees $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_2$ with $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_3$ and a reflection of $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_2$ which colours the bottom rather than the top vertex on the first level of that tree, we obtain precisely $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_4$.

Similarly, the staged trees $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_5$, $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_6$, $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_7$ and $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})_8$ share the same tree graph with different stage structure. Here, again by Corollary 2.9, $\mathbb{P}_8$ is a submodel of $\mathbb{P}_5$, $\mathbb{P}_6$ and $\mathbb{P}_7$ because the stages of the representations of the latter three models are also contained

17 See again Proposition A.1 in the appendix.

18 This is a very well-informed guess, based on a translation of the result in Corollary 2.9 into operations such as the union and intersection of stage sets of staged trees. We omit a proof here because of the rather technical notation and because the algebraic ansatz we follow in this section is more than sufficient to prove our claims.

in the representation of the first. In analogy to the conjecture above, we now suggest that $\mathbb{P}_8$ might be the result of an intersection of the other three models, $\mathbb{P}_5 \cap \mathbb{P}_6 \cap \mathbb{P}_7$.

By Theorem 1, the models $\mathbb{P}_1, \ldots, \mathbb{P}_8$ above are solution sets of the following systems of equations

$$0 = 0 \tag{2.40.1}$$

$$\pi_1(\pi_3 + \pi_4) = \pi_2(\pi_1 + \pi_2) \tag{2.40.2}$$

$$\pi_1\pi_4 = \pi_2\pi_3 \tag{2.40.3}$$

$$\pi_1\pi_4 = \pi_2\pi_3,\ \pi_1(\pi_3 + \pi_4) = \pi_2(\pi_1 + \pi_2) \text{ and } (\pi_1 + \pi_2)\pi_4 = \pi_3(\pi_3 + \pi_4) \tag{2.40.4}$$

$$(\pi_1 + \pi_2)\pi_4 = \pi_3(\pi_1 + \pi_2 + \pi_3) \tag{2.40.5}$$

$$\pi_1\pi_3 = \pi_2(\pi_1 + \pi_2) \tag{2.40.6}$$

$$\pi_1\pi_4 = \pi_2(\pi_1 + \pi_2 + \pi_3) \tag{2.40.7}$$

$$\pi_1\pi_4 = \pi_2(\pi_1 + \pi_2 + \pi_3),\ \pi_1\pi_3 = \pi_2(\pi_1 + \pi_2) \text{ and } (\pi_1 + \pi_2 + \pi_3)\pi_3 = (\pi_1 + \pi_2)\pi_4 \tag{2.40.8}$$

together with the constraint that the indeterminates here are atomic probabilities, so sum to unity, $\pi_1+\pi_2+\pi_3+\pi_4 = 1$, and are positive $\pi_j \in (0, 1)$ for all $j = 1, 2, 3, 4$. The numbering above again identifies the respective staged tree model $\mathbb{P}_i$ with an equation (2.40.i), for $i = 1, \ldots, 8$. We have hereby numbered the indeterminates such that $\pi_1, \ldots, \pi_4$ are the atomic probabilities read from top to bottom in all staged trees in Fig. 2.5.

Naturally, the staged trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_4$ and $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_8$ with three vertices in the same stage—depicted in Figs. 2.5.4 and 2.5.8, respectively—are characterised by three equations each: one for identifying every pair of vertices. However, in both (2.40.4) and (2.40.8) the latter two equations imply the first. So two equations are sufficient to capture these assumptions:

$$\pi_1(\pi_3 + \pi_4) = \pi_2(\pi_1 + \pi_2) \text{ and } (\pi_1 + \pi_2)\pi_4 = \pi_3(\pi_3 + \pi_4) \tag{2.40.4 revisited}$$

$$\pi_1\pi_3 = \pi_2(\pi_1 + \pi_2) \text{ and } (\pi_1 + \pi_2 + \pi_3)\pi_3 = (\pi_1 + \pi_2)\pi_4 \tag{2.40.8 revisited}$$

This is an interesting result because it tells us firstly that our specified system of equations is not minimal—so some of the odds-ratio equations we state are redundant—and that if we specify these equations cleverly then two algebraic expressions can code three graphical assumptions. Again, here this is merely an observation but a future stream of research might lead into an exploration of such a system with nice algebraic properties: for instance by calculating Gröbner bases of ideals generated by the odds-ratio equations above. Doing this for one example, we obtain the additional insight that the equations characterising $\mathbb{P}_4$ are equivalent to a system which includes the assertion that $\pi_2 = \pi_3$ must be true. This is striking because in

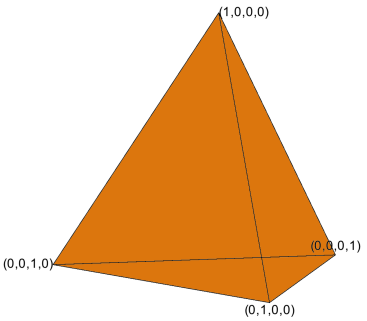
fact whatever parametrisation we choose for the two paths representing these atoms, we will indeed always find that these two atomic probabilities have to be equal: see also (2.39.4). This result was not immediately obvious from the odds-ratio equations and neither from the stage constraints.

So what are the advantages of describing the models $\mathbb{P}_1, \ldots, \mathbb{P}_8$ by (2.40) rather than (2.39)?

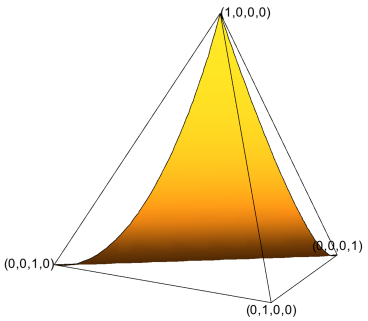
First, it is now easy to check whether or not a given distribution over four atoms belongs to a certain model, or indeed to check which of the four staged trees offers a possible description of a given dataset. For instance, assume we have estimated that a distribution $\boldsymbol{p} = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ which states that every one of the outcomes is equally likely accurately models a given problem. Then plugging the numbers $(\pi_1, \pi_2, \pi_3, \pi_4) = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ into (2.40), we find that the first four equations are true whereas the latter four are false. As a consequence, the given distribution is an element of the models $\mathbb{P}_1, \ldots, \mathbb{P}_4$ but not of the models $\mathbb{P}_5, \ldots, \mathbb{P}_8$. So if we were to choose a staged tree in order to describe our observation, this tree should have two levels rather than three. We can furthermore check that there are distributions over four atoms which cannot be described by staged tree models (except for the saturated model, of course). For instance, $\boldsymbol{p}' = (\frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \frac{4}{10})$ does not fulfil any of the equations in (2.40).

Second, because none of the equations in (2.40) are equivalent to each other, we deduce that all models $\mathbb{P}_1, \ldots, \mathbb{P}_8$ specified above must be different. This can be verified again using `CoCoA`. As a consequence, all staged trees in Fig. 2.5 are representations of pairwise different staged tree models. This conclusion is the algebraic counterpart to a constructive result we elaborate in Section 3.2: we show in that section how to transform a given staged tree into a different staged tree which still represents the same model, using a parametric approach. We can then show that none of the staged trees in Fig. 2.5 are in that sense transformable into each other.

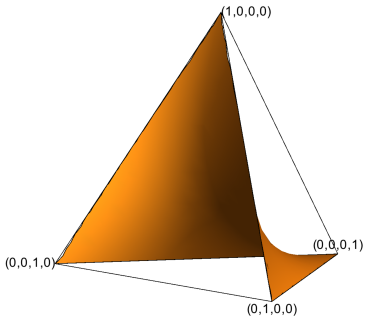
Third, we can now also verify our conjectures above about the relationship between the models $\mathbb{P}_1, \ldots, \mathbb{P}_8$. Indeed, the seven models $\mathbb{P}_2, \ldots, \mathbb{P}_8$ are all submodels of the saturated model $\mathbb{P}_1 = \Delta^\circ_{4-1}$ because every point that fulfils (2.40.2) to (2.40.8) trivially fulfils (2.40.1). In the same fashion, note that (2.40.2) and (2.40.3) together with a renaming of the variables in (2.40.2) yields precisely the three equations in (2.40.4). Hence, we can deduce that the equality $\mathbb{P}_4 = \mathbb{P}_2 \cap \mathbb{P}_3 \cap \mathbb{P}_2^{\text{rotated}}$ is indeed true where $\mathbb{P}_2^{\text{rotated}}$ simply denotes $\mathbb{P}_2$ with permuted components of points. More precisely, because of our result that $\mathbb{P}_4$ is the solution set of only two equations in (2.40.4 revisited), we actually have that $\mathbb{P}_4 = \mathbb{P}_2 \cap \mathbb{P}_2^{\text{rotated}}$ is sufficient and that $\mathbb{P}_3 \supseteq \mathbb{P}_4$. Similarly, the system in (2.40.8) is equivalent to the equations in (2.40.5) to (2.40.7). As a result, the corresponding models $\mathbb{P}_8 = \mathbb{P}_5 \cap \mathbb{P}_6 \cap \mathbb{P}_7$ are intersections of each other. Again, because (2.40.8 revisited) is sufficient to describe this model, we can simplify this to $\mathbb{P}_8 = \mathbb{P}_5 \cap \mathbb{P}_6 \subseteq \mathbb{P}_7$.



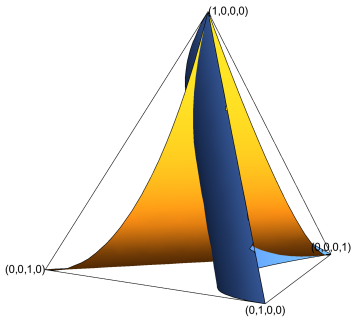
(2.6.1) The saturated model $\mathbb{P}_1 = \Delta^{\circ}_{4-1}$ from (2.39.1) and (2.40.1).



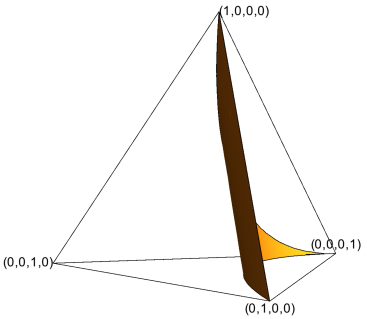
(2.6.2) The staged tree model $\mathbb{P}_2$ from (2.39.2) and (2.40.2).



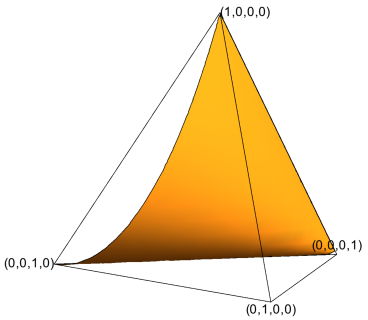
(2.6.3) The staged tree model $\mathbb{P}_3$ from (2.39.3) and (2.40.3).



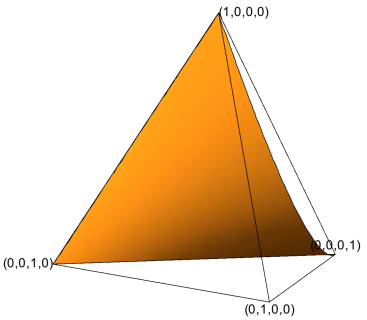
(2.6.4) The staged tree model $\mathbb{P}_4 = \mathbb{P}_2 \cap \mathbb{P}_2^{\text{rotated}}$ from (2.39.4) and (2.40.4).



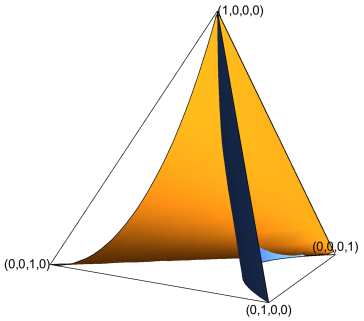
(2.6.5) The staged tree model $\mathbb{P}_5$ from (2.39.5) and (2.40.5).



(2.6.6) The staged tree model $\mathbb{P}_6$ from (2.39.6) and (2.40.6).



(2.6.7) The staged tree model $\mathbb{P}_7$ from (2.39.7) and (2.40.7).



(2.6.8) The staged tree model $\mathbb{P}_8 = \mathbb{P}_5 \cap \mathbb{P}_6$ from (2.39.8) and (2.40.8).

Figure 2.6. All staged tree models on four atoms.

As a final step in this section, we will now provide illustrations for the models above. Recall first that the probability simplex $\Delta^{\circ}_{4-1} \subseteq \mathbb{R}^4$ of which these models are subsets is three-dimensional because the sum-to-one condition $\pi_1 + \pi_2 + \pi_3 + \pi_4 = 1$ defines a hyperplane in $\mathbb{R}^4$ of dimension three. Projecting the solution sets of (2.40) together with the open hypercube $(0,1)^4$ onto that hyperplane, we then obtain objects inside the open three-dimensional tetrahedron Δ°_{4-1} with vertices $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ and $(0,0,0,1)$. Every equation in (2.40) lowers the dimension of the image of that projection by 1: so the saturated model in (2.40.1) is three-dimensional and fills the probability simplex, the staged tree models in (2.40.2), (2.40.3) and (2.40.5) to (2.40.7) are two-dimensional surfaces inside the simplex, and the solution sets of (2.40.4 revisited) and (2.40.8 revisited) are curves within the simplex.

Illustrations of these semi-algebraic sets can then be obtained using symbolic computer programmes such as `Mathematica` (Wolfram Research Inc., 2016). We show the plots provided by this software in Fig. 2.6 where again the depiction in Fig. 2.6.i corresponds to the model $\mathbb{P}_i$ represented by the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_i$ in Fig. 2.5.i for all $i = 1, \ldots, 8$.

Judging naïvely by these figures, all staged tree models on four atoms seem rather smooth algebraic surfaces. This is however not necessarily the case if we plot them outside the simplex constraints. To illustrate this point, consider Fig. 2.7. We depict here the model $\mathbb{P}_6$ from (2.39.6) and (2.40.6), projected onto the hyperplane in $\mathbb{R}^4$ where $\pi_4 = 0$ and ignoring sum-to-1 conditions and positivity constraints. This variety $\mathbf{V}(\pi_3\pi_1 - \pi_2(\pi_1 + \pi_2))$ is a two-sheeted hyperboloid which exhibits a singularity at the origin.

Although the algebra underlying the approach we present here allows for a much deeper analysis, we will in this thesis content ourselves with the rather straightforward insights we have obtained above and with using the obtained illustrations mainly to convey an idea of what staged tree models may look like. An analysis of the more general algebraic and geometric properties of these sets—such as smoothness, dimension, singularities, behaviour on the boundary and the role played by sum-to-1 conditions—remains an open and highly intriguing field of research. An exploration of these properties is especially promising because staged tree models are huge generalisations of Bayesian networks, and the usefulness of algebraic tools in the analysis of these models has already been shown by other authors. So we have here been able to develop the basis for providing great generalisations of well-known results like Proposition 2.10.

In addition, and more central to this thesis, we can deduce from the results above that two staged trees represent the same model if and only if they specify equivalent systems of odds-ratio equations: see also Example 2.8. This of course is the case if and only if both give rise to the same semi-algebraic sets as illustrated for instance in Fig. 2.6. This result is the geometric

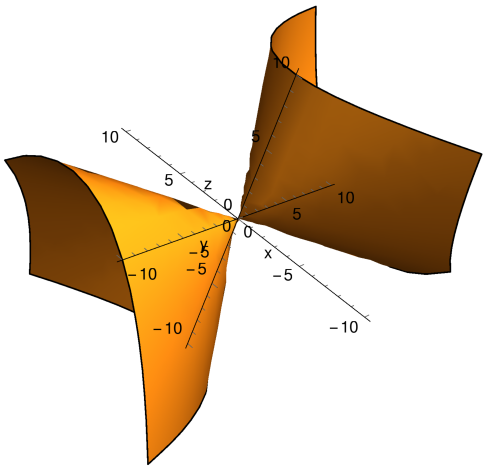


Figure 2.7. When relaxing positivity and sum-to-1 constraints, the staged tree model $\mathbb{P}_6$ from (2.39.6) and (2.40.6) becomes a variety $\mathbf{V}(\pi_3\pi_1 - \pi_2(\pi_1+\pi_2)) \subseteq \mathbb{R}^3$. Compare also the true model in Fig. 2.6.6.

counterpart to the theory we develop in Section 3.2 where we present the tools needed to check whether or not two staged trees are in the above sense statistically equivalent. Centrally, these tools are based on an understanding of polynomials associated to staged trees but will permit us to use graphical rather than geometric representations in order to determine statistical equivalence.

3. The interpolating polynomial

The work in this chapter centres around a formal polynomial in the edge labels of a probability tree, as defined below. We will show the use of this polynomial in two important applications: in calculating marginal and conditional probabilities in staged trees and in classifying staged trees which represent the same families of probability distributions.

We first show that the probabilities of a number of interesting events can be easily inferred from their graphical representation. For instance, the probability of a vertex-centred event equals the probability of arriving at that vertex, and the probability of an edge-centred event equals the probability of arriving at that edge and then passing along it. These calculations involve only very specific labels of the tree graph which play a similarly specific role in the polynomial as greatest common divisors of selected terms. Computations of interest can hence be easily performed via differentiation operations on these terms.

Using the same polynomial, we then show that all staged trees representing the same model either share a common tree parametrisation—and hence have the same edge labels, possibly depicted in a different order—or have different tree parametrisations which however are linked via simple transformations of their associated graphs. So there is a very straightforward connection between a polynomial and a graphical representation of a tree model.

The following definition will be used throughout this chapter.

Definition 3.1 (Network and interpolating polynomial)**.** Let $\mathbb{P}_\Psi \subseteq \Delta^\circ_{n-1}$ be a parametric statistical model on a finite space Ω and with a monomial parametrisation $\Psi : \boldsymbol{\theta} \mapsto \boldsymbol{p}_{\boldsymbol{\theta}} \in \mathbb{P}_\Psi$ for $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^d$. A *network polynomial* for this model is of the form

$$c_{g,\Psi}(\boldsymbol{\theta}) = \sum_{\omega \in \Omega} g(\omega) p_{\boldsymbol{\theta}}(\omega) \tag{3.1}$$

where $g : \Omega \to \mathbb{R}$ is a function assigning real values to atoms. If $g = 1$, then the formal sum $c_{1,\Psi}$ of all atomic probabilities is called an *interpolating polynomial* for $\mathbb{P}_\Psi$.

The idea of associating a polynomial to a graph, or more generally to a parametric statistical model, is not as abstract as it might at first appear. In fact, there are three very intuitive reasons for this approach.

First, inference in statistical models often relies on calculations based on evaluating polynomials and rational functions. In particular, joint probabilities are calculated by multiplying conditional probabilities, conditional probabilities can be calculated using fractions, and marginal probabilities are calculated by summing over joint probabilities. So a thorough analysis of the polynomials involved in these types of expressions promises to give us a better understanding of the mechanisms underlying a model. We show below that the network polynomial is particularly useful in tackling this challenge because it is by definition just a sum of probabilities together with a weighting. In fact, Darwiche (2003) based his approach to inference in Bayesian networks on the idea that the interpolating polynomial with all coefficients set to one and the network polynomials where the function $g = \mathbb{1}_A$ is always an indicator of events $A \in \sigma(\Omega)$ are simply symbolic ways of representing the probabilities of certain events. These can be straightforwardly used for answering probabilistic queries which are of polynomial or rational form.

Second, Pistone et al. (2001b) have shown that a network polynomial as above is simply a generalisation of the concept of a *generating function* for representing a discrete statistical model[19]. The authors have amongst other results been successful in using these to translate inference in statistical models into an algebraic framework and to calculate moments of discrete random variables. Because generating functions can be used to identify certain classes of models, this suggests that network polynomials of probability tree models might be used to generate distributions which factorise according to such a tree graph. This is indeed the case as we will discover below.

Third, computer scientists frequently use acyclic digraphs or tree graphs whose inner nodes correspond to summation or multiplication operations for representing and evaluating values of polynomials. These *arithmetic circuits* or *sum-product nets* are fast inferential tools (Darwiche, 2009). So there is a straightforward and natural link between network polynomials and graphs, our graphical models.

In the first section below, we will embed the network polynomial of a staged tree model in a differential framework, using partial derivatives to answer a collection of probabilistic queries. We then establish a link to the computer science literature to give an idea of how the differential framework—apart from being another elegant representation—benefits computation in staged tree models. In the second section, we will embed the interpolating polynomial into an algebraic framework, using it as a representation of an equivalence class of staged trees.

19 For instance, if $\Omega = \mathbb{Z}_{\geq 0}$ is the state space of a random variable X with probability mass function p, then choosing $g(\omega) = t^\omega$ for some $t \in [-1, 1]$ yields the network polynomial $\sum_{\omega=0}^{\infty} t^\omega p(\omega) = \mathbb{E}[t^X]$, so the *probability generating function* of X (Blitzstein and Hwang, 2014). Note however that in this thesis we always assume Ω to be finite and generically not directly associated to a random variable.

We hereby enrich the polynomial–graph dictionary by a new entry: nested representations of these polynomials are in one-to-one correspondence with labelled event trees. So in that sense, the interpolating polynomial of a staged tree is indeed a (graph-)*generating* polynomial.

3.1. A differential approach

The network polynomial as defined above is simply a sum over all atomic probabilities, multiplying each of these monomial terms by a real factor. So of course when these factors are all equal to unity, then the interpolating polynomial itself sums to one. In a symbolic framework—in close analogy to an algebraic framework—we will not plug in these sum-to-1 conditions but think of the network or interpolating polynomial as a formal sum of indeterminates which can at the very end of our analysis be assigned numerical values. This *symbolic approach* has already been successful in Bayesian network models (Castillo et al., 1995; Chan and Darwiche, 2002; Darwiche, 2003) and can now be extended to staged tree models and more general parametric statistical models.

We will present the main results of Görgen et al. (2015) below and extend these by a number of new examples and additional insights. The follow-up work we have published in Leonelli et al. (2015) provides far more general results and extends the concepts we present here to parametric models with multilinear parametrisations which might not be straightforwardly represented by a staged tree. In this framework, the propositions below are applied to sensitivity analyses where small variations on single (or sometimes also on a collection of) parameters can have severe impacts on the specification of a model. Because these topics are not the focus of this work, the reader is directly referred to the above publication for further reference.

Our first result is that the network polynomial can be used to calculate probabilities of events of interest in any parametric statistical model.

Lemma 3.2 (Network polynomials as functions)**.** *Let $\mathbb{P}_\Psi$ be a parametric model, $\Psi : \Theta \to \mathbb{P}_\Psi$ a monomial parametrisation and let $P_{\boldsymbol{\theta}}(A) = \sum_{\omega \in A} p_{\boldsymbol{\theta}}(\omega)$ denote a probability measure for which $p_{\boldsymbol{\theta}} \in \mathbb{P}_\Psi$, $\boldsymbol{\theta} \in \Theta$ and where $A \subseteq \Omega$ is an event in the underlying space. Denote by $\mathbb{1}_A$ the indicator function of that event, so $\mathbb{1}_A(\omega) = 1$ if $\omega \in A$ and 0 otherwise. Then the network polynomial $c_{g,\Psi}$ with $g = \mathbb{1}_A$ is a function*

$$\begin{aligned} c_{\mathbb{1}_\cdot,\Psi} \;:\; \sigma(\Omega) \times \Theta &\to [0,1] \\ (A, \boldsymbol{\theta}) &\mapsto P_{\boldsymbol{\theta}}(A) \end{aligned} \tag{3.2}$$

which maps an event and a choice of parameters to the probability of that event.

Proof. By Definition 3.1,

$$c_{\mathbb{1}_A,\Psi}(\boldsymbol{\theta}) = \sum_{\omega\in\Omega} \mathbb{1}_A(\omega) p_{\boldsymbol{\theta}}(\omega) = \sum_{\omega\in A} p_{\boldsymbol{\theta}}(\omega) = P_{\boldsymbol{\theta}}(A) \tag{3.3}$$

for any event $A \in \sigma(\Omega)$. □

The less general version of this result concerning Bayesian networks has been previously proven by Darwiche (2003) and the analogous statement concerning staged tree models appears in Görgen et al. (2015). With a view on the previous chapter, note that the lemma above simply states that when we pass from a symbolic framework to one where we assign values to indeterminates—so when using the evaluation homomorphism from Section 1.3.1—then the network polynomial is in fact just a sum of probabilities.

Let now the parametric model in Definition 3.1 always be a staged tree model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$ represented by $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})$ and with graph $\mathcal{T} = (V,E)$. We write

$$c_{g,\mathcal{T}}(\boldsymbol{\theta}) = \sum_{\lambda\in\Lambda(\mathcal{T})} g(\lambda)\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) \tag{3.4}$$

rather than $c_{g,\Psi_\mathcal{T}}$ for the network polynomial of the staged tree in order to avoid double indices. First observe that the interpolating polynomial of a staged tree is square-free if and only if the staged tree is square-free as in Definition 1.11. In more generality, the polynomials of models which have a multilinear monomial parametrisation are always square-free. We will restrict all of the results below to the case of square-free polynomials and staged trees, as done in the original publication.

Also note that within a symbolic framework we will use $\pi_{\boldsymbol{\theta},\mathcal{T}}$ again to denote a symbolic product of edge labels and $P_{\boldsymbol{\theta}}$ for the numerical evaluation of these as in Lemma 3.2 above. We hence need to change back and forth between events, or sets of root-to-leaf paths, $\Lambda \subseteq \Lambda(\mathcal{T})$ represented by a tree and their counterparts $A \subseteq \Omega$ coding the meaning of these events in the context of the respective model. So we will make excessive use of the tree embedding $\iota_\mathcal{T} : A \mapsto \Lambda$ from (1.15) which identifies a graphical representation with such a meaning. Whilst this technicality ensures that the notation in the following equations is unambiguous, we are aware that at a first glance it can also make results hard to read. This drawback should be mitigated by a high number of examples.

Example 3.3 (Calculating probabilities)**.** Return to the developments in a cell culture analysed in Example 1.12 from page 26. Here, the interpolating polynomial of the staged tree $(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})$

from Fig. 1.4.2 is given by

$$\begin{aligned} c_{1,\mathcal{T}}(\boldsymbol{\theta}) &= \theta_{01}\theta_{13}\theta_{37} + \theta_{01}\theta_{13}\theta_{38}\theta_{8,11} + \theta_{01}\theta_{13}\theta_{38}\theta_{8,12} + \theta_{01}\theta_{14}\theta_{49} \\ &\quad + \theta_{01}\theta_{14}\theta_{4,10}\theta_{10,13} + \theta_{01}\theta_{14}\theta_{4,10}\theta_{10,14} + \theta_{02}\theta_{25} + \theta_{02}\theta_{26} \\ &= \theta_{01}\theta_{13}\theta_{37} + \theta_{01}\theta_{13}\theta_{38}\theta_{8,11} + \theta_{01}\theta_{13}\theta_{38}\theta_{8,12} + \theta_{01}\theta_{14}\theta_{37} \\ &\quad + \theta_{01}\theta_{14}\theta_{38}\theta_{8,11} + \theta_{01}\theta_{14}\theta_{4,10}\theta_{8,12} + \theta_{02}\theta_{13} + \theta_{02}\theta_{14} \end{aligned} \tag{3.5}$$

where $\boldsymbol{\theta}$ is the vector of labels $\theta_{ij} = \theta(e_{ij})$ of all edges $e_{ij} = (v_i, v_j)$ depicted in the tree graph. Those edges emanating from vertices in the same stage have been identified in the second equation above. Now, suppose we are interested in calculating the probability of death of a cell in this setting. This event is given by the union of all root-to-leaf paths going through an edge labelled 'die', so the union of edge-centred events $\Lambda(e_{37}) \cup \Lambda(e_{49})$. Then

$$P_{\boldsymbol{\theta}}(\text{death}) = c_{\mathbb{1}_{\Lambda(e_{37})\cup\Lambda(e_{49})},\mathcal{T}}(\boldsymbol{\theta}) = \theta_{01}\theta_{13}\theta_{37} + \theta_{01}\theta_{14}\theta_{37}. \tag{3.6}$$

Here, this coincides precisely with summing all terms in (3.5) which include the label $\theta_{37} = \theta_{49}$.

We will now for the first time apply Darwiche's differential approach to staged tree models and use differentiation operations on the network polynomial in order to infer results as in the example above. We hereby discover that this approach has a natural graphical counterpart and can be much more straightforwardly performed than in Bayesian networks.

Recall first that an edge-centred event $\Lambda(e) \subseteq \Lambda(\mathcal{T})$ contains only root-to-leaf paths passing through its central edge $e \in E$: see Fig. 1.2. So all atomic monomials involved in calculating the probability of that event $\sum_{\lambda\in\Lambda(e)} \prod_{e'\in E(\lambda)} \theta(e')$ are divisible by the indeterminate $\theta(e)$ which labels that edge. Compare also Remark 2.1 and Fig. 2.1 where we used this result in order to cancel out common divisors which were edge labels in a fraction. In addition, because staged trees in this section are always assumed to be square-free, the label $\theta(e)$ appears only once in every term in the polynomial which represents the probability of the event above. Thus, when differentiating the interpolating polynomial of the whole tree with respect to that indeterminate and then multiplying the result by that same indeterminate, $\theta(e)\frac{\partial}{\partial\theta(e)}c_{1,\mathcal{T}}(\boldsymbol{\theta})$, we obtain a sum of all products along root-to-leaf paths which go through an edge labelled $\theta(e)$, and hence the probability of the event 'passing through an edge labelled $\theta(e)$'. Because in staged trees this label is of course often not unique to the edge e, we find that in general this differentiation operation is not sufficient to calculate the probability of interest, so $\theta(e)\frac{\partial}{\partial\theta(e)}c_{1,\mathcal{T}}(\boldsymbol{\theta}) \neq P_{\boldsymbol{\theta}}(\iota_{\mathcal{T}}^{-1}(\Lambda(e)))$.

However, the rationale above does still provide the right intuition for using the interpolating polynomial in answering probabilistic queries of interest. In fact, we can easily circumnavigate the technicality of having non-unique labels by introducing a second, in a sense redundant,

vector of parameters $\boldsymbol{\epsilon} = (\epsilon(e) \mid e \in E)$ whose components can take the values 0 or 1. We think of $\epsilon(e)$ as indicating 'passing along the edge e', so $\epsilon(e) = \mathbb{1}_{\Lambda(e)}$. In this notation,

$$c_{g,\mathcal{T}}(\boldsymbol{\epsilon}, \boldsymbol{\theta}) = \sum_{\lambda \in \Lambda(\mathcal{T})} g(\lambda) \prod_{e \in E(\lambda)} \epsilon(e)\theta(e) \tag{3.7}$$

denotes the network polynomial as in (3.4) together with an extra parameter. Now differentiating this new polynomial (3.7) with respect to one component of the vector $\boldsymbol{\epsilon}$ simply cancels those terms in the network polynomial which are not associated to root-to-leaf paths passing through the associated edge. As a consequence, such a differentiation deletes all probabilities of atoms for which the meaning of that edge is not true. We can think of this operation as setting certain labels in an event tree to zero and cancelling all subtrees emanating from those. A combination of differentiations with respect to $\boldsymbol{\epsilon}$- and $\boldsymbol{\theta}$-components thus enables us to calculate a number of interesting quantities. In particular, we can derive the following results.

Proposition 3.4 (Conditional and joint probabilities)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a probability tree and let $e \in E$ be a fixed edge in the graph $\mathcal{T} = (V, E)$. Let $A(e) = \iota_{\mathcal{T}}^{-1}(\Lambda(e)) \subseteq \Omega$ denote the set of all atoms in the underlying model which are represented by the edge-centred event $\Lambda(e) \subseteq \Lambda(\mathcal{T})$ in the tree, and let $A \subseteq \Omega$ be any other event with positive probability. Then*

$$P_{\boldsymbol{\theta}}(A(e) \mid A) = \frac{1}{c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta})} \frac{\partial c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta}, \boldsymbol{\epsilon})}{\partial \epsilon(e)} \tag{3.8}$$

$$P_{\boldsymbol{\theta}}(A(e) \cap A) = \theta(e) \frac{\partial c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta}, \boldsymbol{\epsilon})}{\partial \theta(e)} \tag{3.9}$$

are the conditional and joint probabilities of these events.

Proof. Lemma 3.2 yields that $P_{\boldsymbol{\theta}}(A) = c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta})$. So if this probability is not equal to zero then by the rule of conditional probability,

$$P_{\boldsymbol{\theta}}(A(e) \mid A) = \frac{P_{\boldsymbol{\theta}}(A(e) \cap A)}{P_{\boldsymbol{\theta}}(A)} = \frac{1}{c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta})} P_{\boldsymbol{\theta}}(A(e) \cap A) \tag{3.10}$$

in the notation above. Further observe that $P_{\boldsymbol{\theta}}(A(e)) = \frac{\partial}{\partial \epsilon(e)} c_{1,\mathcal{T}}(\boldsymbol{\theta}, \boldsymbol{\epsilon})$ because the partial derivative with respect to $\epsilon(e)$ cancels out every monomial which does not contain the indicator for passing along the edge e. As a consequence, we obtain that

$$P_{\boldsymbol{\theta}}(A(e) \cap A) = \sum_{\lambda \in \iota_{\mathcal{T}}(A) \cap \Lambda(e)} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = \frac{\partial}{\partial \epsilon(e)} c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)}, \mathcal{T}}(\boldsymbol{\theta}, \epsilon). \tag{3.11}$$

This proves the claim. □

Because in tree graphs every vertex is in one-to-one correspondence with its unique entering edge, the results of Proposition 3.4 can be naturally modified for calculating conditional and joint events involving vertex-centred rather than edge-centred events: simply by substituting $\Lambda(e) = \Lambda(v')$ for $e = (v, v')$.

Consider now an illustration below.

Example 3.5 (Example 3.3 continued.)**.** In (3.8) and (3.9) above set $A =$ 'death' again to be the event that a unit dies, and let $e = (v_3, v_7)$ be the edge that those units pass along who find themselves dying in a hostile environment with high cell activity. Then Proposition 3.4 yields that

$$P_{\boldsymbol{\theta}}(A(e) \mid \text{death}) = \frac{1}{c_{\iota_{\mathcal{T}}(A),\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})} \frac{\partial c_{\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial \epsilon(e_{37})} = \frac{\theta_{01}\theta_{13}\theta_{37}}{\theta_{01}\theta_{13}\theta_{37} + \theta_{01}\theta_{14}\theta_{37}}, \tag{3.12}$$

$$P_{\boldsymbol{\theta}}(A(e) \cap \text{death}) = \theta_{37} \frac{\partial c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial \theta_{37}} = \theta_{01}\theta_{13}\theta_{37}. \tag{3.13}$$

We omit the $\boldsymbol{\epsilon}$ indeterminates in the solution above for the sake of simplicity. So the probability of a unit dying in that particular situation is simply the fraction of the probability of passing precisely along the root-to-leaf path $(e_{01}, e_{13}, e_{37}) = \iota_{\mathcal{T}}(\text{hostile}, \text{high}, \text{die})$ over the probability of the event 'death'. The joint probability of these events is the probability of death itself, because one event is contained in the other, $A(e_{37}) \subseteq A$.

The probabilities of more general events can now be calculated using higher order derivatives as below.

Corollary 3.6 (Combined expressions)**.** *In the notation of Proposition 3.4, we can calculate the following quantities*

$$P_{\boldsymbol{\theta}}(A(e_1) \cap A(e_2) \mid A) = \frac{1}{c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})} \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial \epsilon(e_1)\partial \epsilon(e_2)} \tag{3.14}$$

$$P_{\boldsymbol{\theta}}(A(e_1) \cap A(e_2) \cap A) = \theta(e_1)\theta(e_2) \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial \theta(e_1)\theta(e_2)} \tag{3.15}$$

$$P_{\boldsymbol{\theta}}(A \mid A(e)) = \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial \theta(e)\partial \epsilon(e)} \tag{3.16}$$

using partial derivatives of the network polynomial.

The proof of this result goes in direct analogy to the proof of Proposition 3.4 above.

Example 3.7 (Example 3.5 continued)**.** We are now interested in calculating the probability of survival and full recovery, given that a unit found itself in a hostile environment. Let thus A = 'hostile' with $\iota_{\mathcal{T}}(A) = \Lambda(v_1)$ and $e_1 = e_{38}$ and $e_2 = e_{8,11}$ in Corollary 3.6. Then, again omitting the $\boldsymbol{\epsilon}$ indeterminates in the solution, we obtain that

$$P_{\boldsymbol{\theta}}(A(e_{38}) \cap A(e_{8,11}) \mid \text{hostile}) = \frac{1}{c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})} \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial\epsilon(e_{38})\partial\epsilon(e_{8,11})}$$
$$= \frac{\theta_{01}\theta_{13}\theta_{38}\theta_{8,11}}{\theta_{01}} = \theta_{13}\theta_{38}\theta_{8,11} \tag{3.17}$$

$$P_{\boldsymbol{\theta}}(A(e_{38}) \cap A(e_{8,11}) \cap \text{hostile}) = \theta_{38}\theta_{8,11} \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial\theta_{38}\partial\theta_{8,11}} = \theta_{38}\theta_{8,11}\theta_{01}\theta_{13} \tag{3.18}$$

$$P_{\boldsymbol{\theta}}(\text{hostile} \mid A(e_{38})) = \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta},\boldsymbol{\epsilon})}{\partial\theta_{38}\partial\epsilon(e_{38})} = \theta_{01}\theta_{13}\theta_{8,11} \tag{3.19}$$

from (3.14) to (3.16), respectively. So the probability of survival and full recovery of a cell in a hostile environment is simply given by the product of primitive probabilities along the edges labelled 'survival, full recovery' on the subpaths emerging from the vertex v_1, which in turn represented the event $\Lambda(v_1)$ = 'hostile environment'. The probability of the joint event 'hostile environment, survival and full recovery' is the probability of the associated root-to-leaf path $(e_{01}, e_{13}, e_{38}, e_{8,11})$. And the probability of the environment being hostile given that we have observed survival is simply the probability of the subpath 'hostile, high activity, full recovery' of $(e_{01}, e_{13}, e_{8,11})$ without the edge e_{38} labelled 'survival'.

The above results would in general be wrong in a framework where we a priori put in sum-to-1 conditions. This is because clearly a partial derivative of the form $\frac{\partial}{\partial\theta_1}\theta_1\theta_2 = \theta_2$ is very different from $\frac{\partial}{\partial\theta_1}\theta_1(1-\theta_1) = 2\theta_1 - 1$ even if $\theta_1 + \theta_2 = 1$. For the same reason, partial derivatives in non square-free staged trees are often meaningless, or at best hard to interpret.

We conclude this section with a brief presentation of how the computations above can be easily performed using the following graphical structure:

Definition 3.8 (Definition 12.2 of Darwiche (2009))**.** An *arithmetic circuit* is a rooted acyclic digraph whose leaves, so vertices without children, are labelled by the indeterminates in an interpolating polynomial (3.7) and whose inner nodes are given by summation + or multiplication × operations.

Figure 3.1 shows an arithmetic circuit for our running example.

Darwiche originally built arithmetic circuits for Bayesian networks in order to provide a compact representation of the probability distributions of the underlying model. Then the

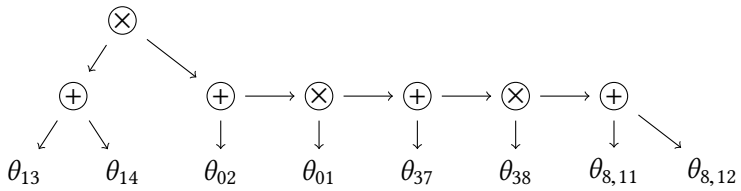


Figure 3.1. An arithmetic circuit for computing the interpolating polynomial of the staged tree from Examples 3.3, 3.5, 3.7 and 3.9.

values of these distributions could be easily calculated when consecutively passing from the leaves to the root-node of the circuit and performing the operations indicated by vertices on the way. Arithmetic circuits are thus alternative representations of labelled event trees, and by the development in the subsequent section both are graphical representations for a polynomial. In Darwiche (2003), arithmetic circuits for Bayesian networks are often also tree graphs. Because in staged trees we know a priori about equality of certain labels, arithmetic circuits representing interpolating polynomials of staged trees are more generally acyclic digraphs. An example of this was provided in Görgen et al. (2015).

Consider an illustration below.

Example 3.9 (Example 3.7 continued)**.** The interpolating polynomial (3.5) of the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ can be written in the simplified form

$$c_{1,\mathcal{T}}(\boldsymbol{\theta}) = (\theta_{13} + \theta_{14})(\theta_{01}(\theta_{37} + \theta_{38}(\theta_{8,11} + \theta_{8,12})) + \theta_{02}). \tag{3.20}$$

This bracketing allows for a straightforward compilation of that polynomial into an arithmetic circuit: shown in Fig. 3.1.

The bracketing of the polynomial in the example above can be inferred directly from the graph of a labelled event tree: see Remark 3.15 and Alg. 1 in the following two sections. For Bayesian networks, Darwiche (2009) provides an algorithm for constructing an arithmetic circuit for a given BN or a given probability distribution. This graph can then be used for evidence propagation, for differentiating a polynomial as we have done above and for determining the computational complexity of evaluating that polynomial. Interestingly, for staged trees, the labelled event tree can be used directly as a vehicle for constructing that circuit: without the need to construct a join tree of a given BN or to employ more complex algorithms as has been necessary in the framework of Darwiche (2003).

So we have been able to show that the differential approach to probabilistic queries in BN models as established by Darwiche translates seamlessly into the framework of staged trees.

Not only are our models more general—so we have significantly extended the value of these results—but differentiations also have very straightforward interpretations in tree graphs. In Section 4.1, we will extend the results above to applications in causal inference.

3.2. Polynomial and statistical equivalence

We will now show the use of interpolating polynomials in an algebraic framework for identifying graphical representations of the same statistical model. In particular, we prove that the interpolating polynomial can fully characterise the class of all staged tree representations of a given tree model.

The study of these statistical equivalence classes is an important one. The first reason for this is computational: staged trees and CEGs constitute a massive space to explore when selecting a model fitted to data. By identifying a single representative within an equivalence class of model representations and a priori selecting across these representatives rather than the full class, we can dramatically reduce the search effort across this space. The second reason concerns coherence: when adopting a Bayesian approach in model selection, two statistically equivalent model representations should be given the same prior distribution over their parameters. To apply this principle, it is essential to know when two staged trees make the same distributional assertions. The third reason is inferential: just as for a Bayesian network, a CEG or staged tree has a natural causal extension, as discussed in Chapter 4. So, in particular, causal discovery algorithms can be applied to staged trees to elicit a putative causal ordering between various associated variables. Clearly a necessary condition for a causal deduction to be made is that this deduction is invariant to the choice of one representative within a statistical equivalence class. So again we need to be able to identify equivalence classes of a hypothesized causal CEG in order to perform these algorithms. More detail on these motivations can be found in Smith et al. (2017).

All the results in this section have now been presented in adapted form in Görgen and Smith (2015). The presentation given in this thesis is however much slower than what has been reported in that publication and has been enriched by a large number of extra illustrations and results.

So let in the following $c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{\lambda \in \Lambda(\mathcal{T})} \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda)$ denote the interpolating polynomial of a staged tree. $c_{\mathcal{T}}$ is a polynomial in the ring $\mathbb{R}[\Theta_{\mathcal{T}}] = \mathbb{R}[\theta(e) \mid e \in E]$ whose indeterminates are edge labels. To make the distinction between the algebraic framework in this section and a setting where the $\theta(e)$, $e \in E$, are probabilities, we will again use the symbol $\pi_{\boldsymbol{\theta},\mathcal{T}}$ only in this formal sense.

With the help of the interpolating polynomial, we will formulate conditions below which are equivalent to the following definition, in the notation from Definitions 1.1 and 1.7.

Definition 3.10 (Statistical equivalence)**.** Two parametrisations Ψ and Φ of the same parametric model $\mathbb{P}_\Psi = \mathbb{P}_\Phi$ are called *statistically equivalent.*

For staged tree models, two representations $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are called *statistically equivalent staged trees* if and only if their tree parametrisations induce the same model, so if and only if $\mathbb{P}_{(\mathcal{S},\boldsymbol{\theta}_\mathcal{S})} = \mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$. The symbol $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ then denotes the set of all staged tree representations of $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_\mathcal{T})}$.

The very general notion of statistical equivalence in the definition above enables us to state that for instance a staged tree and its corresponding CEG, or some alternative representation in form of an acyclic digraph, are statistically equivalent whenever they code the same assumptions over a set of probability distributions: see also Proposition 1.16.

In Bayesian network models, it is common to call two acyclic digraphs $\mathcal{D}$ and $\mathcal{D}'$ *Markov-equivalent* (Lauritzen, 1996) or *distribution-equivalent* (Heckerman, 1998) if their vertices correspond to a common set of random variables and if the conditional independence assumptions over these variables as read from the two graphs are equivalent. They then represent the same set of probability distributions. Andersson et al. (1997) provide a full characterisation of the set $[\mathcal{D}]$ of all acyclic digraphs representing the same BN model. In particular, they find that all elements in $[\mathcal{D}]$ share a strong graphical properties: they have the same *essential graph.* The essential graph is a mixed graph whose undirected edges are inherited from an acyclic digraph within the equivalence class and which includes directed edges only if two vertices in that graph have a common child but are not connected by an edge. So this mixed graph uniquely identifies an equivalence class of acyclic digraphs.

A complete characterisation of the analogous set $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ for staged tree models is the aim of this section. Thwaites and Smith (2015b) have first attempted to solve this problem in a graphical fashion. However, unlike for BN models, two statistically equivalent staged trees have very few graphical properties in common and cannot be elegantly characterised using a surrogate to the essential graph: see a first example below. So instead we here use a different strategy. In particular, we will specify a *polynomial* representation which will uniquely identify a class of statistically equivalent staged trees.

We have seen in Chapter 2 and in particular in Theorem 1 that two staged trees are statistically equivalent if and only if they give rise to equivalent systems of polynomial equations and inequalities. So in order to check for statistical equivalence, it would be straightforward to just follow the strategies outlined in the previous chapter and employ again computer algebra software such as `CoCoA` to determine whether or not two staged trees determine ideals

with the same sets of generators, for instance using Gröbner-basis techniques. However, we will in this section take a more constructive approach. Our strategy in this development is to find a way to derive all different tree parametrisations of the same model, so the elements in the equally coloured clouds around a model in Fig. 0.2. We do this by first characterising those trees which induce the same tree parametrisation—so are in the same cloudy shape in that figure—and then defining an operator which graphically traverses the class of these, so a subclass of $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]$. This is the focus of Section 3.2.1. In Section 3.2.2 we then define a reparametrisation operator which enables us to traverse the class of all graphical representations of the same model—changing from one cloud to the next in that figure. The combination of these two operators enables us to traverse the whole statistical equivalence class of representations of a staged tree model.

Observe first that by definition, two staged trees are statistically equivalent if and only if identified atoms have the same atomic probabilities in both representations.

Example 3.11 (Saturated model)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a saturated probability tree with $n \in \mathbb{N}$ root-to-leaf paths and let $\pi_{\boldsymbol{\theta},\mathcal{T}}$ denote a distribution which factorises according to $\mathcal{T}$. Then $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \Delta^{\circ}_{n-1}$ is a saturated tree model as in Section 2.3.

Let now $\mathcal{F} = (v_0, \{e_1, \ldots, e_n\})$ be a floret with n edges and an associated vector of labels $\boldsymbol{\theta}_{\mathcal{F}} = \big(\theta(e_i) \mid i = 1, \ldots, n\big) \in \Delta^{\circ}_{n-1}$. Then the *star* tree graph which is the labelled floret $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ is a probability tree.

By definition, the star is statistically equivalent to the saturated tree if and only if the probabilities associated with the same atoms are identified. So $\theta(e_i) = \pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda_i)$ for all edges and paths $\iota^{-1}_{\mathcal{F}}(e_i) = \iota^{-1}_{\mathcal{T}}(\lambda_i)$ which have the same meaning, $i = 1, \ldots, n$. In this case, both models naturally coincide as sets: also $\mathbb{P}_{(\mathcal{F},\boldsymbol{\theta}_{\mathcal{F}})} = \Delta^{\circ}_{n-1}$.

Because of this statistical equivalence, a star can be thought of as a graphically *minimal* representation of a saturated model.

3.2.1. The swap operator

We first characterise a class of staged trees for which the interpolating polynomial is invariant in the formal algebraic framework from Section 1.3. This class can be traversed using a simple local operation which is based on algebraic and graphical properties of certain subgraphs.

Importantly, in this algebraic framework we will again refrain from a priori imposing sum-to-1 conditions on the florets of a probability tree and from substituting values for primitive probabilities. The result in Proposition 1.6 justifies our approach of imposing these conditions at the very end of every tree transformation: this is because local summing-to-unity conditions imposed on the florets of a labelled event tree do not violate global summing-to-one of

atomic probabilities. It is only this global condition which has to be unchanged between two representations of the same model.

Definition 3.12 (Polynomial equivalence)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ be two staged trees with the same underlying space Ω. We say that $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ are *polynomially equivalent* if and only if their network polynomials coincide as objects of the same ring, $c_{g,\mathcal{S}} = c_{g,\mathcal{T}} \in \mathbb{R}[\Theta_{\mathcal{T}}]$ for all polynomials $g \in \mathbb{R}[\Theta_{\mathcal{T}}]$.

We immediately see that when two network polynomials are equal for any polynomial g, they are also termwise equal and the corresponding atomic probabilities can be identified. So we have the following:

Lemma 3.13 (Sufficiency)**.** *Polynomially equivalent staged trees are statistically equivalent.*

Proof. Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ be polynomially equivalent. Set $g_\omega = \mathbb{1}_{\{\omega\}}$ to be the indicator function[20] of an arbitrary element $\omega \in \Omega$ in the underlying discrete space. By assumption, the network polynomials $c_{g_\omega,\mathcal{S}} = c_{g_\omega,\mathcal{T}}$ are equal. Hence,

$$\pi_{\boldsymbol{\theta},\mathcal{S}}(\iota_{\mathcal{S}}(\omega)) = c_{\mathbb{1}_{\{\omega\}},\mathcal{S}}(\boldsymbol{\theta}) = c_{\mathbb{1}_{\{\omega\}},\mathcal{T}}(\boldsymbol{\theta}) = \pi_{\boldsymbol{\theta},\mathcal{T}}(\iota_{\mathcal{T}}(\omega)) \tag{3.21}$$

for all $\omega \in \Omega$. Thus, the atomic probabilities coincide pairwise and therefore, by the evaluation homomorphism, the distributions $\pi_{\boldsymbol{\theta},\mathcal{S}} = \pi_{\boldsymbol{\theta},\mathcal{T}}$ are equal. This implies that $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ are statistically equivalent. □

By the lemma above, two staged trees which are polynomially equivalent have the same interpolating polynomial and we can unambiguously identify the terms (atomic monomials) in the polynomial with the respective atoms. Centrally, in square-free staged trees, this can be achieved by inverting the symbolic function $\pi_{\boldsymbol{\theta},\mathcal{T}} : \lambda \mapsto \prod_{e \in E(\lambda)} \theta(e)$ which is injective[21]. This identification is important in any application of these models where we would not wish to assume that a tree model is invariant to rotations within the probability simplex. For instance, if in Example 1.14 the outcome that a child is 'poor and ill' was very likely with 'rich and ill' being very unlikely, then we would not want to allow a permutation of these probabilities: compare also our motivation for introducing the tree embedding in (1.15). However, we can nonetheless formally analyse the properties of polynomially equivalent staged trees without referring to an underlying space. This follows our approach in Chapter 2 and in particular in Section 2.2 where

20 By Pistone et al. (2001a), indicator functions are indeed polynomials. So these can be analysed as objects in polynomial rings just like in the previous chapter.

21 In the non square-free staged tree from Fig. 2.5.4, the atomic probabilities $\pi_2 = \pi_3$ are always equal. So the corresponding tree parametrisation is not injective. We prove injectiveness in the square-free case in the appendix: see Proposition A.2.

we identified staged tree models as sets of distributions (or points) irrespective of which values of the distribution were assigned to which atom. Compare also the *equivariance principle* of Casella and Berger (2002). We will thus henceforth denote the class of polynomially equivalent staged trees as the subset $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]^c \subseteq [\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ of statistically equivalent staged trees which share a common interpolating polynomial $c = c_\mathcal{T}$.

All trees within such a class give rise to ideals and varieties with the same parametrisations, so within the same rings and with the same elimination framework as set up in Section 2.2.

In general, polynomial equivalence is not necessary for statistical equivalence: see for instance the saturated probability tree in Example 3.11 where every non-star representation would have a different parametrisation and hence a different interpolating polynomial. Examples 3.35 and 3.39 below present interesting cases where polynomial and statistical equivalence are equivalent. In X-compatible staged trees, we can think of polynomially equivalent trees as a set of graphical representations of a model which share the same monomial representation in terms of potentials (see Remark 1.9) with different local normalisations on florets. Example 3.30 below illustrates this point in more detail.

Example 3.14 (Polynomial equivalence)**.** Consider the staged trees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ in Figs. 3.2.1 and 3.2.2, respectively, ignoring for the moment their thick depicted subtrees. Both staged trees have the same interpolating polynomial

$$c_\mathcal{T}(\boldsymbol{\theta}) = \theta_1 + \theta_2\theta_4 + \theta_2\theta_5 + \theta_3\theta_4\theta_6 + \theta_3\theta_4\theta_7 + \theta_3\theta_4\theta_8 + \theta_3\theta_5 = c_\mathcal{S}(\boldsymbol{\theta}) \tag{3.22}$$

in the edge labels $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_8)$.

So the atomic monomials are the same across both representations but the respective tree parametrisations read these in a different order. For instance, $\Psi_{\mathcal{T},4}(\boldsymbol{\theta}) = \theta_3\theta_4\theta_6$ and $\Psi_{\mathcal{S},4}(\boldsymbol{\theta}) = \theta_4\theta_3\theta_6$ for the fourth root-to-leaf path counting from top to bottom in Fig. 3.2. In a commutative framework, the two parametrisations are of course equal $\Psi_\mathcal{T} = \Psi_\mathcal{S}$ because $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are polynomially equivalent. Compare Remark 3.18 for an interpretation of this observation.

Note further that $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ have different sum-to-1 conditions on their florets: either $\theta_0 + \theta_1 + \theta_2 = 1$ and $\theta_3 + \theta_4 = 1$, or $\theta_0 + \theta_3 + \theta_4 = 1$ and $\theta_1 + \theta_2 = 1$. These four equations cannot simultaneously hold if the tree labels are primitive probabilities, so strictly positive. However, it is still easy to show that in both graphs the respective local floret sum-to-1 conditions $\sum_{e \in E(v)} \theta(e) = 1$ for all $v \in V$ imply that also atomic probabilities sum to unity, $\sum_\lambda \prod_{e \in E(\lambda)} \theta(e) = 1$. This was proven in Proposition 1.6. As a consequence, local sum-to-1 conditions between different polynomially equivalent staged trees can vary while leaving the atomic probabilities invariant.

The remark below has been central to this new development of determining statistical equivalence and will now pave the way for constructively traversing a class of polynomially equivalent staged trees.

Remark 3.15 (Nested factorisation). By Definition 3.1, the interpolating polynomial of a staged tree is simply a sum over all atomic monomials, calculated as the product of all edge labels along a root-to-leaf path in the graph $\mathcal{T} = (V, E)$.

Centrally, the tree graph also yields a way to parenthesise the interpolating polynomial as follows. For every floret $\mathcal{F}_v$ where $v \in V$ is the parent of a leaf, we sum all components of its vector of labels $\boldsymbol{\theta}_v$ and multiply the result by its parent label $\theta(\mathrm{pa}(v), v)$. We then sum the result over the parent's labels $\boldsymbol{\theta}_{\mathrm{pa}(v)}$. By repeating this until all labels are summed and we arrived at the root $\mathrm{pa}(v) = v_0$, the interpolating polynomial can then be written in terms of a nested factorisation:

$$c_{\mathcal{T}}(\boldsymbol{\theta}) \;=\; \sum_{v_1 \in \mathrm{ch}(v_0)} \theta(v_0, v_1) \Big(\sum_{v_2 \in \mathrm{ch}(v_1)} \theta(v_1, v_2) \;\ldots\; \Big(\sum_{v_k \in \mathrm{ch}(v_{k-1})} \theta(v_{k-1}, v_k) \Big) \Big). \tag{3.23}$$

Here, the index $k \in \mathbb{N}$ in every innermost bracket implicitly depends on the length of a root-to-leaf path $((v_0, v_1), (v_1, v_2)), \ldots, (v_{k-1}, v_k))$, and can be different for every such sequence of edges.

In (3.23), every inner bracket includes a sum over the children of a vertex whose parent's children are summed in an outer bracket. In particular, the innermost brackets correspond to a sum over labels of a leaf- and the outermost to the labels of the root-floret. This bracketing thus inductively follows the unfolding of paths in the underlying event tree.

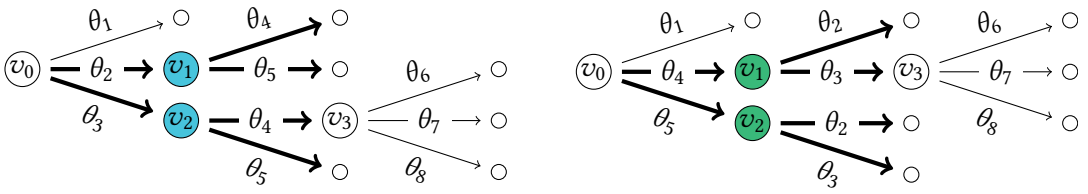
Example 3.16 (Example 3.14 continued.)**.** The staged trees from Example 3.14 induce two different orders of bracketing of their interpolating polynomials, namely:

$$c_{\mathcal{T}}(\boldsymbol{\theta}) = \theta_1 + \theta_2(\theta_4 + \theta_5) + \theta_3(\theta_4(\theta_6 + \theta_7 + \theta_8) + \theta_5) \tag{3.24.1}$$

$$c_{\mathcal{S}}(\boldsymbol{\theta}) = \theta_1 + \theta_4(\theta_2 + \theta_3(\theta_6 + \theta_7 + \theta_8)) + \theta_5(\theta_2 + \theta_3) \tag{3.24.2}$$

where again $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_8)$. Just like in (3.23), we can see in (3.24.1) and (3.24.2) which labels are attached to the root in the two different trees and which are attached to downstream edges and leaves in Figs. 3.2.1 and 3.2.2, respectively.

So a nested representation of the interpolating polynomial of a staged tree seems to be strongly linked to the graph of that tree. Using the generalisation of this idea below, we can prove the strength and usefulness of that link in Proposition 3.19.



(3.2.1) $(\mathcal{T}, \theta_{\mathcal{T}})$ **(3.2.2)** $(\mathcal{S}, \theta_{\mathcal{S}})$

Figure 3.2. Two staged trees $(\mathcal{T}, \theta_{\mathcal{T}})$ and $(\mathcal{S}, \theta_{\mathcal{S}})$ which share the same interpolating polynomial: see Examples 3.14 and 3.16. The thick depicted subtrees are twins $(\mathcal{T}, \theta_{\mathcal{T}})_u$ and $(\mathcal{S}, \theta_{\mathcal{S}})_u$ in the stage $u = \{v_1, v_2\}$, and the map $\mathfrak{s} : (\mathcal{T}, \theta_{\mathcal{T}}) \mapsto (\mathcal{S}, \theta_{\mathcal{S}})$ is a swap: see Example 3.23.

Definition 3.17 (Tree compatibility)**.** Let $\mathbb{P}_\Psi$ be a discrete parametric model with monomial parametrisation Ψ and associated polynomial ring denoted $\mathbb{R}[\Theta] = \mathbb{R}[\theta_1, \ldots, \theta_d]$. We call any polynomial $c \in \mathbb{R}[\Theta]$ in that ring *tree compatible* if it admits a representation of the form

$$c(\boldsymbol{\theta}) \;=\; \sum_{\theta_1 \in A_0} \theta_1 \Big(\sum_{\theta_2 \in A(\theta_1)} \theta_2 \Big(\sum_{\theta_3 \in A(\theta_2)} \theta_3 \;\ldots\; \Big(\sum_{\theta_k \in A(\theta_{k-1})} \theta_k \Big) \Big) \Big) \tag{3.25}$$

where every index set $A_0, A(\theta_{j-1}) \subseteq \{\theta_1, \ldots, \theta_d\}$ is a set of parameters that contains at least two elements, $j \in \{1, \ldots, k\}$. Again, $k \in \mathbb{N}$ implicitly depends on the number of indeterminates in one monomial term of the polynomial. We write $s(c(\boldsymbol{\theta}))$ for one fixed order of summation of the terms in $c(\boldsymbol{\theta})$ as above, and call this a *tree-compatible factorisation* of the polynomial.

In Görgen et al. (2017), we give a recursive definition of tree compatibility which relies on a less cumbersome notation. We will however use the bracketed recursive form here in order to make it easier to see the link between a bracketed polynomial and a tree graph. In particular, we show in Proposition 3.19 below that in fact in a tree-compatible factorisation of a polynomial each index set in (3.25) will be the set of edge labels in one floret in a corresponding tree graph as in (3.23).

Interestingly, this type of recursive bracketing of a polynomial has already been studied in the context of Bayesian networks.

Remark 3.18 (X-compatible factorisations). The interpolating polynomial of every X-compatible tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ admits an explicit tree-compatible factorisation

$$c_{\mathcal{T}}(\boldsymbol{\theta}) \;=\; \sum_{x_{A_1} \in \mathbf{X}_{A_1}} \theta(x_{A_1}) \Big(\sum_{x_{A_2'} \in \mathbf{X}_{A_2'}} \theta(x_{A_2}) \Big(\sum_{x_{A_3'} \in \mathbf{X}_{A_3'}} \theta(x_{A_3}) \ldots \Big(\sum_{x_{A_k'} \in \mathbf{X}_{A_k'}} \theta(x_{A_k}) \Big) \Big) \Big) \tag{3.26}$$

where the index sets A_i are as in (1.17) and we use the shorthand $A_i' = A_i \setminus A^{i-1}$ for summing indices without repetition, $i \geq 2$. Lauritzen and Spiegelhalter (1988) observe that the recursive

nesting of sums in (3.26) provides a very efficient way to compute joint probabilities from marginals in a BN model. However, the authors do not observe that this is entirely due to the fact that every discrete Bayesian network can be represented by a staged tree and that every probability tree admits such a bracketing. So using our representation of a BN model, the computationally advantageous factorisation above comes for free by (3.23).

By Proposition 3.19 below, the tree embedding (1.17) of an underlying state space into an X-compatible staged tree with the above polynomial can be recovered from (3.26). Every factorisation like the one above then reads the underlying monomial parametrisation $\Psi_{\mathcal{T}}$ in a non-commutative way as in Example 3.14. This non-commutativity corresponds precisely to an interpretation of the potentials above as respective marginal or conditional probabilities, depending on their order of multiplication. We will see examples of this below. So different tree-compatible factorisations of the interpolating polynomial of an X-compatible staged tree correspond to different normalisations of the underlying probability mass function: see also Examples 1.10 and 3.29.

An important aspect of Remark 3.15 is that this result is reversible: not only can we easily read a bracketed representation of the interpolating polynomial from a labelled event tree, but we can also construct a tree graph from a tree-compatible factorisation of a polynomial—the use of the term 'tree compatible' is thus justified. In addition, all polynomially equivalent staged trees arise from a tree-compatible reordering of a given order of summation. Each of these gives a different representations within the same statistical equivalence class:

Proposition 3.19 (Probability tree models)**.** *Let $\mathbb{P}_\Psi$ be a discrete parametric model with multilinear monomial parametrisation Ψ and interpolating polynomial $c = c_{1,\Psi} \in \mathbb{R}[\Theta]$. Then Ψ is a tree parametrisation if and only if there exists a tree-compatible factorisation $s(c(\boldsymbol{\theta}))$ of c as in (3.25) for which the conditions that $\theta_i \in (0,1)$ and $\sum_{\theta_{i+1} \in A(\theta_i)} \theta_{i+1} = 1$ for all $i = 1, \dots, d$, are not contradictory.*

In this case, the map $\mathfrak{c} : s(c(\boldsymbol{\theta})) \mapsto (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is invertible and there exists a square-free probability tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with parametrisation $\Psi_{\mathcal{T}} = \Psi$ representing this model, so $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})} = \mathbb{P}_\Psi$.

Proof. Sufficiency of the first part of the claim is straightforward. Indeed, by Definition 3.17 and Remark 3.15, the interpolating polynomial of a tree model is tree compatible, and by Definition 1.7 the parametrisation of a probability tree model is called a tree parametrisation.

For necessity assume now the interpolating polynomial $c = c_{1,\Psi} \in \mathbb{R}[\Theta]$ of the model to be tree compatible and given by the factorisation $s(c(\boldsymbol{\theta}))$ in (3.25). We construct a labelled graph as follows. For every subsum of (3.25), draw a floret $\mathcal{F}_j = (v_j, \{e \mid \theta(e) = \theta_j \in A_j(\theta_{j-1})\})$ with one edge for every indeterminate in the sum and attach the indeterminates as labels to these edges, $j = 1, \dots, k$. Then partially order these florets by reversing the steps in Remark 3.15, such that

θ_j is the parent label of the floret whose attached parameters are $A(\theta_j)$, for all $j = 1, \ldots, k$. So the labels of inner sums in (3.25) appear in subtrees rooted after the edges corresponding to outer sums in that bracketing. In this way, we have constructed a connected graph with no cycles—and hence a tree graph—whose leaf-floret edges are labelled by the innermost factors $A_k(\theta_{k-1})$ of $s(c(\boldsymbol{\theta}))$ and the root's edges by the outermost factors A_1. Since by definition every set $A_j(\theta_{j-1})$ has at least two elements, it follows that there are at least two edges in every floret. So the tree-compatible factorisations in (3.23) and (3.25) are componentwise equal and, multiplying out the brackets of $s(c(\boldsymbol{\theta}))$, we find that $c = c_{\mathcal{T}}$. Thus, we have constructed a labelled event tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with the given interpolating polynomial. Because positivity and local sum-to-1 conditions are assumed to be non-contradictory, Proposition 1.6 implies that any such choice will yield a model $\mathbb{P}_\Psi \subseteq \Delta^\circ_{n-1}$. So $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is a probability tree which represents the model $\mathbb{P}_\Psi = \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$, and Ψ is a tree parametrisation.

By construction, the steps above are reversible. As a consequence, the map $\mathfrak{c}$ which identifies a tree-compatible factorisation with a labelled event tree is invertible. □

The above proposition provides us with a powerful tool to decide if a parametric model is a (probability) tree model. In particular, we immediately find:

Corollary 3.20 (Staged tree models)**.** *In the setting of Proposition 3.19, the tree representation $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ of a parametric model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})} = \mathbb{P}_\Psi$ is staged only if all constraints on the parameters are of the form $A_{i+1}(\theta_i) = A_{j+1}(\theta_j)$ for some $i \neq j$ in the notation of Definition 3.17.*

Now, the result above provokes two natural questions. First, *how can we decide whether or not a given interpolating polynomial is tree compatible?* We have answered this question in Görgen et al. (2017) and will present the key ideas in Section 3.3 below.

The second question is: *how do we infer all possible orders of bracketing of an interpolating polynomial $c_{\mathcal{T}}$ which is known*[22] *to be tree compatible?* If we are able to do this, then, using the map $\mathfrak{c}$ in Proposition 3.19 and the construction outlined in the proof, we can obtain all tree representations in the polynomial equivalence class $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c \subseteq [\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]$. So we next study how to reorder the bracketing in a given tree-compatible factorisations.

Clearly, a transformation between two tree-compatible factorisations of the same interpolating polynomial is a change in the order of the summation and bracketing of edge labels. It is thus an application of the distributive property of addition and multiplication in the ring $(\mathbb{R}[\Theta_{\mathcal{T}}], +, \cdot)$. Using Proposition 3.19, we show below that a map between polynomially equi-

22 Note that in applications of staged tree models we are often given a model representation, for instance inferred from a dataset. Using the methods we develop here, we can then—via the polynomial associated to that representation—draw out all alternative model representations.

valent staged trees can be characterised by this application of + and · and can equivalently also be characterised by a finite number of corresponding intuitive graph transformations.

In order to introduce these graph transformations, we first need to define the type of structure these local changes act on:

Definition 3.21 (Twins)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree. We call a subtree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ a *twin* if all of its root-to-leaf paths are sequences of exactly two edges, its root has exactly two children both of which are in the same stage u and there are no other elements in that stage in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$.

So twins are subtrees where one parent-vertex has two identical children florets.

Figure 2.2 shows two staged trees which are twins, and the thick depicted subtrees of the staged trees in Fig. 3.2 are also twins. We provide further examples below.

By construction, the interpolating polynomial of a twin $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ equals

$$c_{\mathcal{T}_u}(\boldsymbol{\theta}) = \Big(\sum_{e \in E(v_0)} \theta(e)\Big)\Big(\sum_{e' \in E(u)} \theta(e')\Big) \tag{3.27}$$

where v_0 denotes the root of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$ and $E(u)$ denotes the edge set of one representative of the stage $u = \mathrm{ch}(v_0)$ in the twin. So whenever we have this type of graph structure, we obtain—after projecting on the subgraph—an interpolating polynomial which factorises into two sums of edge labels. This polynomial admits at least two different tree-compatible factorisations: one whose root-labels are giving by indeterminates in the subsum $\sum_{e \in E(v_0)} \theta(e)$, and one whose root-labels are given by those in $\sum_{e' \in E(u)} \theta(e')$. So by Proposition 3.19 there is a staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_u$ polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$ which reverses its order of depicting events. In terms of the original staged tree, there is a polynomially equivalent labelled event tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ which coincides with $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ everywhere except on the twin.

The map between these two different polynomially equivalent (sub-)trees provides precisely the local transformation we have been looking for:

Definition 3.22 (Swap)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree and let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a twin with stage u. Denote by $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_u \subseteq (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ the staged tree which is polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u$ as in the notation above.

We call the transformation $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ a *naïve swap*. If $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ is a staged tree then $\mathfrak{s}$ is called a *swap*.

So a swap is an operation on a staged tree which allows us to replace a certain two-level subtree of the original tree by one which is polynomially equivalent to this one. This local intervention leaves the remaining staged tree invariant.

Example 3.23 (Example 3.16 continued.)**.** The thick depicted subtrees in Fig. 3.2 are twins $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ in Fig. 3.2.1 and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u \subseteq (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ in Fig. 3.2.2. We can see here that the map $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \to (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ does indeed 'swap' the order of edges before and after the stage $u = \{v_1, v_2\}$ which contains precisely two children of the root.

In terms of the associated tree-compatible factorisations, we find in analogy to the factorisation (3.27) above that the interpolating polynomial of these twins can be written as

$$(\theta_2 + \theta_3)(\theta_4 + \theta_5) = \theta_2(\theta_4 + \theta_5) + \theta_3(\theta_4 + \theta_5) \tag{3.28.1}$$

$$= \theta_4(\theta_2 + \theta_3) + \theta_5(\theta_2 + \theta_3) \tag{3.28.2}$$

where the recursive bracketings in (3.28.1) and (3.28.2) are in one-to-one correspondence with $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})_u$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})_u$, respectively.

We distinguish swaps and naïve swaps because if the reversal of a subtree is performed in a 'naïve' way then the resulting labelled event tree might not be staged. This is because there are polynomials which admit a tree-compatible factorisation of the form (3.25) but violate the assumptions of Proposition 3.19 and Corollary 3.20. These graphs often have single edge labels across different florets identified with each other rather than full vectors of floret labels. This can then lead to contradictory sum-to-1 conditions, and as a consequence these labelled event trees might not even be probability trees. We will see an example of this below. So while both staged trees and otherwise constrained labelled event trees can share the same interpolating polynomial and can be transformed into each other using naïve swaps, they might not always be *staged* tree representations of the same model.

In particular, swaps are well-defined operators $\mathfrak{s} : [\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]^c \to [\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]^c$ within the class of polynomially equivalent staged trees—we see below that they are actually automorphisms of this class—while naïve swaps may map to a codomain larger than the polynomial equivalence class $[\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]^c$ and this codomain may include non-staged trees which are not saturated.

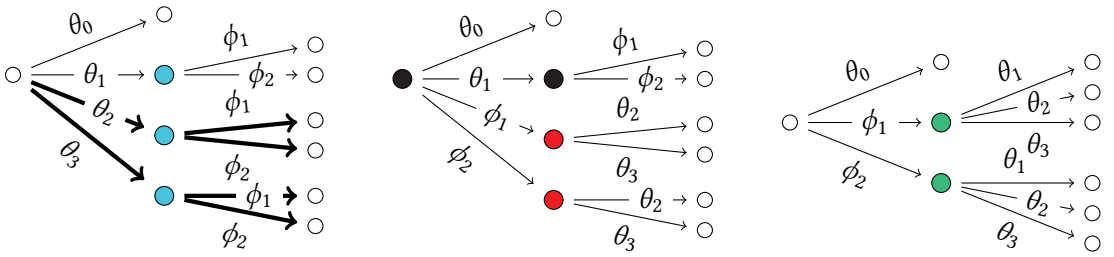
Consider the illustration below.

Example 3.24 (Swaps and naïve swaps)**.** Consider the labelled event trees in Fig. 3.3. All of these three graphs share the edge labels $\theta_0, \theta_1, \theta_2, \theta_3, \phi_1, \phi_2$, and all have the same interpolating polynomial

$$c_\mathcal{T} = \theta_0 + \theta_1(\phi_1 + \phi_2) + \theta_2(\phi_1 + \phi_2) + \theta_3(\phi_1 + \phi_2) \tag{3.29.1}$$

$$c_\mathcal{U} = \theta_0 + \theta_1(\phi_1 + \phi_2) + \phi_1(\theta_2 + \theta_3) + \phi_2(\theta_2 + \theta_3) \tag{3.29.2}$$

$$c_\mathcal{S} = \theta_0 + \phi_1(\theta_1 + \theta_2 + \theta_3) + \phi_2(\theta_1 + \theta_2 + \theta_3) \tag{3.29.3}$$



(3.3.1) A staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

(3.3.2) A non-staged tree probability tree $(\mathcal{U}, \boldsymbol{\theta}_{\mathcal{U}})$.

(3.3.3) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$.

Figure 3.3. Three polynomially equivalent labelled event trees. The maps between them are swaps, naïve swaps and level-swaps. See Example 3.24 and Example 3.26.

here given in the three different tree-compatible factorisations (3.29.i) for Fig. 3.3.i respectively, i = 1, 2, 3.

Denote by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_u \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ the twin which is thick depicted in Fig. 3.3.1. The map $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{U}, \boldsymbol{\theta}_{\mathcal{U}})$ which reorders that twin is a naïve swap. This is because $(\mathcal{U}, \boldsymbol{\theta}_{\mathcal{U}})$ in Fig. 3.3.2 is not a staged tree: the labels ϕ_1 and ϕ_2 occur repeatedly within the graph—in the florets with black coloured vertices—but this edge-label identification is not a stage constraint because it does not identify *vectors* of floret labels as demanded in Definition 1.11. In fact, the two black coloured vertices are not in the same stage but do not have disjoint sets of labels either.

The map which reorders the green twin in Fig. 3.3.3 however is a swap. This operator maps one staged tree onto another, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) \mapsto (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

Now, in both staged trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ the local sum-to-1 conditions on florets imply summing-to-unity of the atomic probabilities. However, on the labelled event tree $(\mathcal{U}, \boldsymbol{\theta}_{\mathcal{U}})$ we would need to impose that both $\phi_1 + \phi_2 = 1$ and $\theta_0 + \theta_1 + \phi_1 + \phi_2 = 1$. This cannot simultaneously be true if these labels are to be positive probabilities. $(\mathcal{U}, \boldsymbol{\theta}_{\mathcal{U}})$ is thus not a probability tree, even if it arises from a tree-compatible factorisation of the interpolating polynomial of a staged tree. Note in addition that while the labelled event trees in Figs. 3.3.1 and 3.3.2 have the same graph $\mathcal{T} = \mathcal{U}$ with differently labelled edges, the graph $\mathcal{S}$ in Fig. 3.3.3 is very different. So polynomial equivalence does not imply that the graphs of two staged trees have to be equal.

The simple example above illustrates the subtleties in defining an algebraic operation on a graphical object: while the polynomial reordering in terms of naïve swaps might not violate a specified set of atomic monomials, its interpretation as a representation of a statistical model might be significantly more complicated, if not impossible.

Proposition 3.25 (The swap operator)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a staged tree. Then $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is a staged tree which is polynomially equivalent to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ if and only if there exists a composition $\mathfrak{s}$ of naïve swaps which form a swap such that $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) = \mathfrak{s}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$.*

Proof. If $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is a staged tree arising from $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ via a composition of naïve swaps, then $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ and $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ have to have shared the same interpolating polynomial. By Proposition 3.19, they are thus polynomially equivalent.

In contrast, let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ be two polynomially equivalent staged trees. Then both have the same interpolating polynomial and different tree-compatible factorisations. The only way in which a subsum

$$\sum_{\theta_j \in A(\theta_{j-1})} \theta_j \left(\sum_{\theta_{j+1} \in A(\theta_j)} \theta_{j+1} \right) \quad + \quad \sum_{\theta'_j \in A(\theta'_{j-1})} \theta'_j \left(\sum_{\theta'_{j+1} \in A(\theta'_j)} \theta'_{j+1} \right) \tag{3.30}$$

can be reordered is by an application of the distributive property when $A(\theta_j) = A(\theta'_j)$. By Proposition 3.19, this is the case if and only if (3.30) is the tree-compatible factorisation of a twin. A composition of these reorderings on the bracketing—and hence a composition of naïve swaps—will thus transform $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ into $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$. □

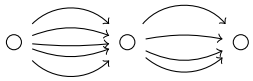
So the swap operator is a simple algebraic and graphical operation which allows us to fully traverse the polynomial equivalence class of a staged tree. In this process, the components of vectors of floret labels $\boldsymbol{\theta}_v$, $v \in V$, might be reordered while the overall set of labels $\{\theta(e) \mid e \in E\}$ remains invariant. We henceforth call a composition of swaps for which vectors of floret labels—and hence local sum-to-1 conditions—are invariant a *floret-swap*. A composition of swaps which permute two levels of a tree is called a *level-swap*.

Example 3.26 (Examples 3.23 and 3.24 continued.)**.** Note that in Example 3.23 the vector of root labels $(\theta_1, \theta_2, \theta_3)$ in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is not present in $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$, and conversely $(\theta_1, \theta_4, \theta_5)$ as depicted in $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is not a vector of floret labels in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. So the swap $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is not a floret swap and $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ have different sum-to-1 conditions. By Lemma 3.13 and Proposition 3.25, both staged trees are representations of the same model. So even if the numerical value of say $\theta_1 = \theta(e_1)$ is different in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$, via a renormalisation it is still the probability of the event $\iota_\mathcal{T}^{-1}(\Lambda(e_1)) \subseteq \Omega$ which is depicted by both trees. The meaning of this parameter is thus the same for all members of a polynomial equivalence class and can be identified across different graphs.

In Example 3.24, the swap $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \to (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ is a level-swap which keeps sum-to-1 conditions invariant. The naïve swap $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{U}, \boldsymbol{\theta}_\mathcal{U})$ is neither a floret- nor a level-swap.

So swaps exchange the order of events in a staged tree. In fact, we find the following intuitive result.

Remark 3.27 (A note on independence). By Proposition 3.19, a factorisation of an interpolating polynomial as in (3.27) above finds its direct graphical counterpart in the twin. In terms of a CEG representation of the same model, this factorisation is even easier to see. Whenever a chain event graph contains a subgraph structure of the following type



we can apply a swap on these two florets.

Interestingly, floret structures as above play a central role in the separation theorems stated in Thwaites and Smith (2015b). In particular, the authors find that edge-centred events $\Lambda(e)$, for edges $e = (v_0, v)$ on the first level of a twin are independent of those $\Lambda(e')$, $e' = (v, v')$, on the second level. This result can also be straightforwardly calculated as presented in Smith et al. (2017). Our very plausible discovery is that for these independent events, the order of $\Lambda(e)$ happening before $\Lambda(e')$ is reversible within the polynomial equivalence class, using the swap operator. So both orders of events are valid representations of the same model. This result will be central to our discussion of a causal interpretation of these classes in Chapter 4.

We have seen above that the polynomial equivalence classes of staged trees can be fully characterised by the local graph transformations given by swaps. This result is not unfamiliar and closely linked to similar results in Bayesian networks. In particular, the swap operator is a close tree analogue of an *arc reversal* in decomposable BN models (Schachter, 1988). These, just like swaps, allow one to traverse the class of all graphical representations of the same model while renormalising but not marginalising the associated probability mass function.

We can in particular state the following result, in the notation from Section 1.2.2.

Corollary 3.28 (Arc reversals)**.** *Let $p_{\boldsymbol{\theta}}(x) = \theta(x_{C_1})\theta(x_{C_2}) \cdots \theta(x_{C_k})$ be the clique-parametrisation of a decomposable BN model as in (1.18). Let $c = \sum_{x \in \mathbf{X}} p_{\boldsymbol{\theta}}(x)$ be the interpolating polynomial of that model in the given parametrisation. Then c is tree compatible and all possible X-compatible staged tree representations of the BN model are members of the polynomial equivalence class $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c$.*

Proof. First, observe that in the interpolating polynomial atomic probabilities can be summed recursively, starting from a sum over the members of an initial clique C_1, and then consecutively

summing over the remaining indices excluding those which have already been included:

$$\begin{aligned} c &= \sum_{x\in\mathbf{X}} p_{\boldsymbol{\theta}}(x) = \sum_{x\in\mathbf{X}} \theta(x_{C_1})\theta(x_{C_2})\cdots\theta(x_{C_k}) \\ &= \sum_{x_{C_1}\in\mathbf{X}_{C_1}} \theta(x_{C_1}) \Big(\sum_{x_{C_2\setminus C_1}\in\mathbf{X}_{C_2\setminus C_1}} \theta(x_{C_2}) \Big(\sum_{x_{C_3\setminus C_1\cap C_2}\in\mathbf{X}_{C_3\setminus C_1\cap C_2}} \theta(x_{C_3}) \quad \cdots \quad \Big(\sum_{x_{C_k\setminus\bigcap_{j=1}^{k-1} C_j}\in\mathbf{X}_{C_k\setminus\bigcap_{j=1}^{k-1} C_j}} \theta(x_{C_k}) \Big) \Big) \Big) \\ &= \sum_{x_{C_1}\in\mathbf{X}_{C_1}} \theta(x_{C_1}) \Big(\sum_{x_{C_2\setminus B_1}\in\mathbf{X}_{C_2\setminus B_1}} \theta(x_{C_2}) \Big(\sum_{x_{C_3\setminus B_2}\in\mathbf{X}_{C_3\setminus B_2}} \theta(x_{C_3}) \quad \cdots \quad \Big(\sum_{x_{C_k\setminus B_{k-1}}\in\mathbf{X}_{C_k\setminus B_{k-1}}} \theta(x_{C_k}) \Big) \Big) \Big) \end{aligned} \tag{3.31}$$

for $x = (x_{C_1}, \ldots, x_{C_k})$ and cliques $C_1, \ldots, C_k$ and separators $B_2, \ldots, B_k$ for $k \in \mathbb{N}$. By Proposition 3.19, there is an X-compatible staged tree induced by the tree-compatible factorisation above. This tree is an alternative graphical representation of the decomposable BN model. In addition, for every permutation of the clique numbering $\{1, \ldots, k\}$, we can find a tree-compatible factorisation as in (3.31) above. Thus, in the polynomial equivalence class $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c$ there is an X-compatible stratified staged tree for every possible numbering of cliques, and every junction tree representation of the BN model. □

So Corollary 3.28 states that for every member of the 'junction tree equivalence class' of the decomposable model[23] there is a corresponding member in the polynomial equivalence class of staged trees. In particular, along every root-to-leaf path in a staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) = \mathfrak{c}(s(c(\boldsymbol{\theta}))$ with summation order s as in (3.31), the probability mass function is read as the product of marginal and conditional probabilities:

$$p(x) = p_{C_1}(x_{C_1}) p_{C_2\setminus B_2}(x_{C_2}|x_{B_2}) \cdots p_{C_k\setminus B_k}(x_{C_k}|x_{B_k}) \qquad \text{for } x \in \mathbf{X}. \tag{3.32}$$

So the potentials in (1.18) are normalised according to the graph structure of the staged tree, just like in Example 1.10. Compare Remarks 1.9 and 3.18 for an early indication of this result. As a consequence, the polynomial equivalence class of a staged tree with clique-parametrisation contains all information we can infer from the BN model.

Example 3.29 (Example 1.10 continued)**.** In the polynomial equivalence class of the X-compatible staged tree whose atomic monomials are given by (1.19), we find one representative for every junction tree representation of the acyclic digraph of Fig. 1.3:

$$c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{x\in\mathbf{X}} \theta(x_{\{1,2,3\}})\theta(x_{\{2,3,4\}})\theta(x_{\{3,5\}})\theta(x_{\{4,6\}}) \tag{3.33.1}$$

$$= \sum_{x_{\{1,2,3\}}} \theta(x_{\{1,2,3\}}) \Big(\sum_{x_{\{4\}}} \theta(x_{\{2,3,4\}}) \Big(\sum_{x_{\{5\}}} \theta(x_{\{3,5\}}) \Big(\sum_{x_{\{6\}}} \theta(x_{\{4,6\}}) \Big) \Big) \Big) \tag{3.33.2}$$

23 See page 22.

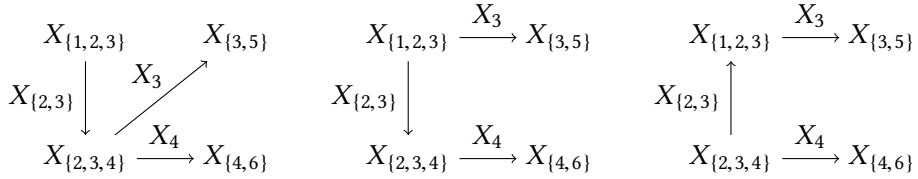


Figure 3.4. Junction tree representations for the model analysed in Example 3.29.

$$= \sum_{x_{\{1,2,3\}}} \theta(x_{\{1,2,3\}}) \Big(\sum_{x_{\{5\}}} \theta(x_{\{3,5\}}) \Big(\sum_{x_{\{4\}}} \theta(x_{\{2,3,4\}}) \Big(\sum_{x_{\{6\}}} \theta(x_{\{4,6\}}) \Big) \Big) \Big) \tag{3.33.3}$$

$$= \sum_{x_{\{2,3,4\}}} \theta(x_{\{2,3,4\}}) \Big(\sum_{x_{\{1\}}} \theta(x_{\{1,2,3\}}) \Big(\sum_{x_{\{5\}}} \theta(x_{\{3,5\}}) \Big(\sum_{x_{\{6\}}} \theta(x_{\{4,6\}}) \Big) \Big) \Big) \tag{3.33.4}$$

$$= \sum_{x_{\{2,3,4\}}} \theta(x_{\{2,3,4\}}) \Big(\sum_{x_{\{5\}}} \theta(x_{\{3,5\}}) \Big(\sum_{x_{\{1\}}} \theta(x_{\{1,2,3\}}) \Big(\sum_{x_{\{6\}}} \theta(x_{\{4,6\}}) \Big) \Big) \Big) \tag{3.33.5}$$

$$= \dots \tag{3.33.6}$$

where the first tree-compatible factorisation corresponds precisely to the tree we constructed in Example 1.10, and the other factorisations correspond to alternative representations of the same model. The factorisation of a probability mass function according to these trees can be read from the tree-compatible factorisation above as outlined in Remark 1.9. For instance, we thus obtain the following renormalisations of the underlying probability mass function

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p_{123}(x_1, x_2, x_3) p_4(x_4|x_2, x_3) p_5(x_5|x_3) p_6(x_6|x_4) \tag{3.34.1}$$

$$= p_{123}(x_1, x_2, x_3) p_5(x_5|x_3) p_4(x_4|x_2, x_3) p_6(x_6|x_4) \tag{3.34.2}$$

$$= p_{234}(x_2, x_3, x_4) p_1(x_1|x_2, x_3) p_5(x_5|x_3) p_6(x_6|x_4) \tag{3.34.3}$$

$$= p_{234}(x_2, x_3, x_4) p_5(x_5|x_3) p_1(x_1|x_2, x_3) p_6(x_6|x_4) \tag{3.34.4}$$

$$= \dots \tag{3.34.5}$$

for $p = p_\theta$ and all $x \in \mathbb{X}$ where (3.34.i) can be read from a root-to-leaf path in the tree with nested factorisation given by (3.33.i+1) for i= 1, . . . , 4.

As a consequence, the first three equations (3.33.1), (3.33.2) and (3.33.3) above correspond precisely to the alternative junction tree representations in Figs. 3.4.1 to 3.4.3, respectively. This is the result of Corollary 3.28.

By the example below, the polynomial equivalence class of staged trees representing a BN model however is much richer than the acyclic digraph equivalence class above.

Example 3.30 (Example 1.18 continued)**.** Consider again the staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 1.5. Its interpolating polynomial $c_{\mathcal{T}}$ admits the following four tree-compatible factorisations, in the notation of Example 1.14.

$$c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{s,e,l=0,1} p_{\boldsymbol{\theta}}(s,e,l) = \sum_{s,e=0,1} \theta(s,e)\Big(\sum_{l=0,1} \theta(s,l)\Big) \tag{3.35.1}$$

$$= \sum_{s,l=0,1} \theta(s,l)\Big(\sum_{e=0,1} \theta(s,e)\Big) \tag{3.35.2}$$

$$= \sum_{l=0,1} \theta(0,l)\Big(\sum_{e=0,1} \theta(0,e)\Big) + \sum_{e=0,1} \theta(1,e)\Big(\sum_{l=0,1} \theta(1,l)\Big) \tag{3.35.3}$$

$$= \sum_{l=0,1} \theta(1,l)\Big(\sum_{e=0,1} \theta(1,e)\Big) + \sum_{e=0,1} \theta(0,e)\Big(\sum_{l=0,1} \theta(0,l)\Big). \tag{3.35.4}$$

We denote the staged trees corresponding to (3.35.1) to (3.35.4) by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_3$, respectively.

Consider Fig. 3.5. From Example 1.18, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is an alternative staged tree representation of a BN model. Because in (3.35.1) we sum over values of S and E first and then vary over L in a potential depending on both S and L, we can label the levels of the corresponding staged tree by the joint random variable (S, E) and the conditional random variable $L|S$.

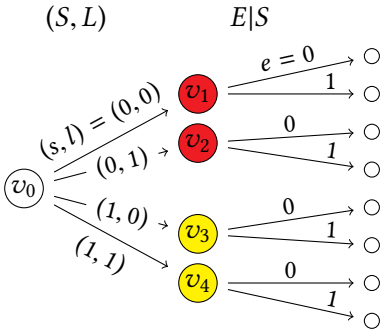
Now, the map $\mathfrak{s}_1 : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$ is a level-swap in both twins in Fig. 1.5. The resulting staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$ in Fig. 3.5.1 then represents the joint variable (S, L) first and $E|S$ last. Since $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$ are in the same polynomial equivalence class, we see immediately that $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c$ is sufficiently rich to contain tree representations which renormalise the probability mass function $\pi_{\boldsymbol{\theta},\mathcal{T}}$ to $\pi_{\boldsymbol{\theta},\mathcal{S}}(\iota_{\mathcal{S}}(s,e,l)) = \theta(s,l)\theta(s,e)$. This illustrates again the result of Corollary 3.28. There are thus at least two different (S,E,L)-compatible representations in $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c$. See also Example 3.39 below.

However, unlike $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$, the staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2 = \mathfrak{s}_2(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 3.5.2 is not (S,E,L)-compatible. This tree is (X,Y)-compatible for new random variables

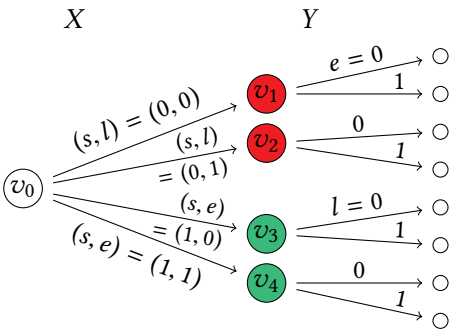
$$X = \begin{cases} (S,L) & \text{if } S=0 \\ (S,E) & \text{if } S=1 \end{cases} \quad \text{and} \quad Y = \begin{cases} E|L & \text{if } S=0 \\ L|E & \text{if } S=1. \end{cases}$$

We call such a transformation which does not only renormalise but redefine problem variables a *twist*. $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_3$ is also a twist of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

The example above provides a very simple illustration of how the statistical equivalence class of a staged tree (or CEG) can be so much larger than the class of Markov-equivalent acyclic digraph representations of the same BN model. It also demonstrates how staged trees



(3.5.1) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$ illustrating an arc reversal.



(3.5.2) A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2$ illustrating a twist.

Figure 3.5. Two staged trees polynomially equivalent to the one in Fig. 1.5. See Example 3.30.

can implicitly generate relationships between new random variables, constructed as functions of the original ones: possibly useful in later interpretative analysis. A more detailed discussion of this process is given in Section 3.2.4. We refer to Thwaites and Smith (2015b) and Smith et al. (2017) for a more technical presentation of how to read random variables from a staged tree.

Concluding this section, we stress the fact that the polynomial equivalence class of a staged tree can be a proper subclass of its statistical equivalence class—and there is thus the need to define a second operator before we are able to fully traverse the whole statistical equivalence class of a given staged tree.

In fact, because saturated probability trees contain no twins, we immediately find:

Corollary 3.31 (Saturated polynomial equivalence)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a saturated probability tree with interpolating polynomial $c = c_{\mathcal{T}}$. Then $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is polynomially equivalent only to itself, so $\#[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]^c = 1$.*

So we cannot apply the swap operator on saturated probability trees or on saturated subtrees of staged trees. Consider an illustration of the impact of this result below.

Example 3.32 (Orders and swaps)**.** Let X, Y be binary random variables with a joint positive probability mass function $p_{\boldsymbol{\theta}}(x, y) = \theta(x)\theta(x, y)$ for $x, y \in \{0, 1\}$. The corresponding discrete model $\mathbb{P}_\Psi$ with parametrisation

$$\Psi : \Big(\theta(x), \theta(x, y) \mid x, y = 0, 1\Big) \mapsto \Big(p_{\boldsymbol{\theta}}(x, y) \mid x, y = 0, 1\Big) \tag{3.36}$$

can be represented by an (X, Y)-compatible staged tree we denote $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. Because we did not impose any constraints on the problem variables, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is saturated. The interpolating

polynomial of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ equals

$$c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{x=0,1} \theta(x) \Big(\sum_{y=0,1} \theta(x,y) \Big) \tag{3.37}$$

given in the corresponding tree-compatible factorisation. We can immediately see the result of Corollary 3.31 in (3.37): there is no second tree-compatible factorisation of $c_{\mathcal{T}}(\boldsymbol{\theta})$. So $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is the only member of its polynomial equivalence class.

This uniqueness might be a spurious limitation in settings where we want to interpret the order of events depicted by $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. In particular, all members of the polynomial equivalence class of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ depict X before Y. However, clearly no such order can be inferred from the model: $\mathbb{P}_{\Psi}$ is also a BN model which can equivalently be represented by the two acyclic digraphs $X \to Y$ and $Y \to X$. So (3.36) can be renormalised to $p_{\boldsymbol{\theta}}(x,y) = \theta(y)\theta(y,x)$ for $x, y = 0, 1$. Hence, in the statistical equivalence class $[\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}]$ of staged tree representations of $\mathbb{P}_{\Psi} = \mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$ there is another (X,Y)-compatible tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) \neq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ which depicts Y before X. Importantly, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ are both saturated and are not polynomially equivalent. There is no swap $\mathfrak{s} : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ and so within the polynomial equivalence class we cannot see that both of these staged trees represent the same model.

So in this section we have found that nested tree-compatible factorisations of interpolating polynomials and labelled event trees are duals of each other: both code the same information and can without loss be transformed into each other. We then discovered that an interpolating polynomial can have more than one such factorisation. In particular, all different tree-compatible factorisations of the same polynomial correspond to labelled event trees with the same atomic monomials. Those of these labelled event trees which are staged trees are then members of what we call a polynomial equivalence class. This class is a subclass of the statistical equivalence class, and there exists a local operation—called a swap—which enables us to transform a staged tree into a polynomially and statistically equivalent staged tree. This operation is able to mimic the behaviour of an arc reversal when a decomposable BN is represented by a staged tree instead.

3.2.2. The resize operator

We now extend the characterisation of polynomial equivalence classes to the corresponding classes of statistically equivalent staged trees—so to all representations of the same model. In order to do this, we need to find sensible ways of reparametrising between the associated polynomial rings $\mathbb{R}[\Theta_{\mathcal{T}}]$ and $\mathbb{R}[\Theta_{\mathcal{S}}]$ of two statistically equivalent trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. Lemma 3.36 presents an interpretation of the term 'sensible' in the context we are interested in,

and Lemma 3.37 completes the extension. Note that it is this reparametrisation which requires us to leave the algebraic framework set up in Sections 1.3.2 and 2.2 and to refrain from the use of elimination ideals if want to answer the question of statistical equivalence constructively.

The second operator we define below is again characterised by local changes which have both an algebraic and a graphical interpretation. Whilst swap operators act on those subtrees which embed conditional independence information—so colour—in a staged tree, the new operator will act on those that either do not provide any information about the model—in terms of a colouring—or that do so in a redundant manner. In particular, we note that for efficient computation it is often helpful to replace uninformative subtrees by graphically simpler florets. This is precisely what the new operator will enable us to do.

We will henceforth call a subtree of a staged tree an *improper* subtree if it is not necessarily an event tree: its root-floret may be a single edge.

Definition 3.33 (Resize)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree and let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})' \subseteq (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be an improper subtree. We denote by $\mathfrak{r}$ the map which transforms $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})'$ into a star $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ whose edge labels are given by the vector $\boldsymbol{\theta}_{\mathcal{F}} = \left(\pi_{\boldsymbol{\theta},\mathcal{T}'}(\lambda') \mid \lambda' \in \Lambda(\mathcal{T}')\right)$ of atomic probabilities in the subtree, and which leaves the remaining staged tree invariant.

We call $\mathfrak{r}$ and its inverse $\mathfrak{r}^{-1}$ *naïve resize* operators, and a *resize* if $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ is a staged tree.

We have seen a first example of resize in Example 3.11 where showed that a saturated tree and a star can represent the same model.

So in general when resizing a staged tree, a subtree or an improper subtree is replaced by a floret while the remainder of the new tree is identical to the original. By construction, atomic probabilities over the root-to-leaf paths of these trees are hereby preserved. Resizes are thus again very local changes which can be applied on certain parts of a given labelled event tree without affecting the remainder of the system. A staged tree and its image under a resize however always have different graphs, so in particular different edges with different labels. They are thus by definition not polynomially equivalent.

Note that while the resize as defined above is effectively a contraction of a subtree, its inverse—also termed a resize—is an expansion of a floret into a subtree with multiple levels. In algebraic terms, a naïve resize thus performs a *substitution* of products of edge labels into monomials of degree 1, whereas its inverse is a rational function. So again as with the swap operator, our development is based here on a duality between algebra—in terms of monomial and rational maps—and graphs—in terms of contraction and expansion of subtrees.

Consider an illustration below.

Example 3.34 (Resizing a floret)**.** The star on the left hand side of Fig. 3.6 is a labelled floret with three edges. We can now apply a resize $\mathfrak{r}$ to this graph in order to transform it into a

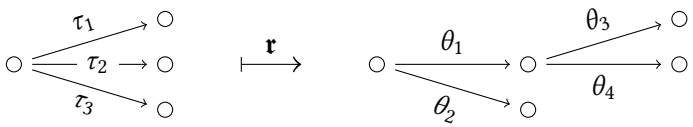


Figure 3.6. A resize of a star with three edges into a binary tree with four edges but three root-to-leaf paths. See Example 3.34.

binary tree with two levels and two florets, depicted on the right hand side of that figure. This resize thus performs an expansion of a labelled event tree into another 'larger' labelled event tree with more edges and vertices but with the same number of atoms.

The labels of these trees under the resize are changed as follows:

$$(\tau_1, \tau_2, \tau_3) \mapsto (\theta_1\theta_3, \theta_1\theta_4, \theta_2) \quad \text{and} \quad \Big(\theta_1, \theta_2, \theta_3, \theta_4\Big) \mapsto (\tau_1 + \tau_2, \tau_3, \frac{\tau_1}{\tau_1 + \tau_2}, \frac{\tau_2}{\tau_1 + \tau_2}\Big) \tag{3.38}$$

such that products of edge labels along root-to-leaf paths which are identified in both representations are the same: as in Definition 3.33 above.

By construction, the star is thus simply labelled by the probabilities of the root-to-leaf paths in the binary tree, and the binary tree's edge labels are calculated based on the rules for joint and conditional probabilities. It is easily checked that sum-to-1 conditions are not violated under this operation. For if $\tau_1 + \tau_2 + \tau_3 = 1$ in the star then also $\theta_1 + \theta_2 = \tau_1 + \tau_2 + \tau_3 = 1$ and $\theta_3 + \theta_4 = \frac{\tau_1}{\tau_1+\tau_2} + \frac{\tau_2}{\tau_1+\tau_2} = 1$ in the binary tree. Conversely, if $\theta_1 + \theta_2 = 1$ and $\theta_3 + \theta_4 = 1$ is true then also $\tau_1 + \tau_2 + \tau_3 = \theta_1(\theta_3 + \theta_4) + \theta_2 = 1$.

Again, just like with the swap operator, a 'naïve' application of this operation can severely violate stage structure. This is because when resizing a staged tree into a single floret, we loose all identifications between its edges. If this information is not redundant then a naïve resize will take us out of the class of staged tree representations for the same model. Consider an illustration below.

Example 3.35 (The collider)**.** Consider a BN given by binary random variables X_1, X_2 and X_3, and a *collider* graph $X_1 \to X_3 \leftarrow X_2$ representing the assumption that X_1 is independent of X_2 (Smith, 2010). We can represent this model by an (X_1, X_2, X_3)-compatible staged tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ where level i corresponds to the random variable X_i, $i = 1, 2, 3$. Then because $X_1 \perp\!\!\!\perp X_2$ are independent, the transition probabilities from X_2 do not depend on X_1. This implies that both children of the root are in the same stage. So the two-level subtree emanating from the root is a twin and can be swapped. The primitive probabilities on the third level however will be pairwise different because X_3 is not (conditionally) independent of any of the other two variables.

The polynomial equivalence class of such a staged tree enables us to level-swap X_1 and X_2, in the same way that we can permute the order of these vertices in an acyclic digraph representation, and keeps X_3 fixed on the third level. When there are no confounders, the order 'X_1 and X_2 happen before X_3' can be interpreted as having a potential chronological or causal meaning: see Chapter 4.

Now, any naïve resize operator on $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ would substitute factors in an atomic monomial by terms of lower degree. Every such substitution would necessarily involve edges emanating from coloured vertices and would hence violate stage information. In particular, these stage constraints would then not be captured graphically anymore and might translate into a set of non-linear equations. Any naïve resize will thus not yield a staged tree as in Definition 1.11 but a tree model analogous to a context-specific BN, namely a graph together with some extra non-graphical assumptions.

The result below establishes various useful criteria under which $\mathfrak{r}$ is a well-defined map between two staged trees representing the same model.

Lemma 3.36 (Non-naïve resizes)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ be a staged tree. A composition $\mathfrak{r}$ of naïve resizes applied to $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is a resize if one of the following conditions is fulfilled:*

a) $\mathfrak{r}$ only acts on saturated (improper) subtrees.
b) $\mathfrak{r}$ only acts on (improper) subtrees which are polynomially equivalent to each other and whose vertices are not in the same stage as vertices not contained in these subtrees.

Proof. a) Because the image $\mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) = (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ of a staged tree under a naïve resize is again a probability tree and since by assumption the non-trivial stage sets of image and preimage coincide, clearly also $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) \in [\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}]$ is a staged tree.

b) The assumptions in this case imply that the stage-structure of the naïvely resized subtrees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})' \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is self-contained in the sense that if we can show that every resized subtree $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})' = \mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'$ is a staged tree, then there are no extra constraints within the original tree $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and its image $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) = \mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ which could violate the model assumptions. Now, because all subtrees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'$, $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'' \subseteq (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ that $\mathfrak{r}$ acts on are polynomially equivalent, we find in $\mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})'$ and $\mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})''$ that the atomic probabilities $\pi_{\boldsymbol{\theta}, \mathcal{T}'}(\lambda') = \pi_{\boldsymbol{\theta}, \mathcal{T}''}(\lambda'')$ coincide formally for subpaths λ', λ'' which have the same atomic monomial in $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$. Thus, the image $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S}) = \mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ is a staged tree where the stages are given by these identified formerly atomic, now primitive labels. □

Note that case (a) in Lemma 3.36 enables us to contract saturated subtrees into florets—these are uninformative to a specification of the model in terms of stage constraints. Case (b)

enables us to directly identify atomic monomials of polynomially equivalent subtrees rather than repeating stage equations edge by edge. So for instance vertices in the same position can have their identified subtrees resized.

From the above, we deduce immediately:

Lemma 3.37 (Sufficiency)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree and $\mathfrak{r}$ a composition of naïve resize operators applied under the conditions of Lemma 3.36. Then $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and the image $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) = \mathfrak{r}(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ are statistically equivalent staged trees.*

We can now employ the resize to overcome restrictions of the polynomial equivalence class.

Example 3.38 (Example 3.32 continued)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be the (X, Y)-compatible probability tree representation of the saturated model $\mathbb{P}_\Psi$ and Ψ the parametrisation in (3.36). Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ depict X before Y and let $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ be the (X, Y)-compatible representation of $\mathbb{P}_\Psi$ which depicts Y before X. We denote by $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ the star representation of the saturated model $\mathbb{P}_\Psi$, defined as in Example 3.11. Then there exist two resizes $\mathfrak{r}_1$ and $\mathfrak{r}_2$ with

$$(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \overset{\mathfrak{r}_1}{\mapsto} (\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}}) \overset{\mathfrak{r}_2}{\mapsto} (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) \tag{3.39}$$

which are not naïve resizes because they act on a saturated tree. These transform one representation into the other. By Lemma 3.37, the three staged trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$, $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}}) = \mathfrak{r}_1(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}}) = \mathfrak{r}_2 \circ \mathfrak{r}_1(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ are statistically equivalent.

Just like the swap operator found its analogon in the arc reversal, also resize operations have well-known counterparts in Bayesian network models. We illustrate this in an example below and find that, in particular, clever applications of resizes can enable us to restrict our analysis of a model to sufficiently expressive polynomial equivalence classes.

Example 3.39 (Resizes in BN models)**.** Note that in Example 1.14 we represented the decomposable graph $S \to E \to L$ by a staged tree with two levels whose labels were actually associated to joint and conditional random variables, so vertices in a graph $(S, E) \to (E, L)$. The latter is an acyclic digraph whose vertices are the cliques of the original graph—so a junction tree. Performing this operation in the representation of a BN model changes any associated probability mass function into a clique-parametrisation (1.18). So applying a resize on a staged tree like this enables us to restrict the analysis of the statistical equivalence class of that tree to a polynomial equivalence class which contains all the information the BN could provide: see Corollary 3.28. In this sense, resizes of saturated subtrees can be thought of as providing a 'minimally sufficient' parametrisation of the model.

Of course the other way around the resize operator can not only be used to obtain a most compact graphical description of a model but also to obtain the most detailed one. Following the result of Example 3.34 above, we can indeed show that every staged tree is statistically equivalent to a *binary* staged tree where every floret has precisely two edges.

The existence of this 'maximal' description of an underlying model will enables us in Section 3.3 to find bounds on the size of a statistical equivalence class of staged trees.

We have thus introduced a second operator which has a dual effect on the algebraic and the graphical representation of a given staged tree model and which enables us to take a staged tree out of its polynomial equivalence class while staying inside the same statistical equivalence class. This procedure is analogous to choosing a clique-parametrisation for a decomposable Bayesian network in order to avoid redundant graphical structure.

3.2.3. The full statistical equivalence class

The swap and the resize operator when used in conjunction are sufficiently powerful to enable us to traverse the whole equivalence class of a given staged tree. Both make local changes on a graph which are justified by analogous local operations on the interpolating polynomial, namely rebracketing and substitution operations. We have seen in the former two sections that naïve swaps and naïve resizes do not change a given probability distribution over a set of atoms, so the resulting graph is always a probability tree representation of the underlying model—even though it is not necessarily staged. Similarly, compositions of these naïve operators can still result in well-defined operators between staged trees, as presented above. So restricting these naïve operators to those which in composition yield an object in the class of staged trees, we obtain our main result:

Theorem 2 (Statistical equivalence)**.** *Two staged trees $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are statistically equivalent if and only if there exists a map $\mathfrak{m} : (\mathcal{T}, \boldsymbol{\theta}_\mathcal{T}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ which is a finite composition of swaps and resizes.*

Proof. Let $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ be statistically equivalent staged trees. Then the atomic probabilities $\pi_{\boldsymbol{\theta},\mathcal{T}}(\lambda) = P_{\boldsymbol{\theta}}(\iota_\mathcal{T}^{-1}(\lambda)) = \pi_{\boldsymbol{\theta}',\mathcal{S}}(\lambda')$ of identified root-to-leaf paths $\lambda' = \iota_\mathcal{S}(\iota_\mathcal{T}^{-1}(\lambda))$ are always equal. Here, $P_{\boldsymbol{\theta}}$ denotes again the underlying measure on Ω that $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ have in common. In order to prove existence of a map $\mathfrak{m}$ between these two trees, we distinguish two cases:

First, if the above equality of atomic probabilities holds in a formal sense for every $\lambda \in \Lambda(\mathcal{T})$ then $(\mathcal{T}, \boldsymbol{\theta}_\mathcal{T})$ and $(\mathcal{S}, \boldsymbol{\theta}_\mathcal{S})$ are polynomially equivalent. In this case, Lemma 3.13 states that a map exists between the two staged trees which is a composition of swaps, and thus proves the claim.

Second, if formal equality is not the case then the two staged trees have different parametrisations. In this case denote by $\Lambda \subseteq \Lambda(\mathcal{T})$ the set of root-to-leaf paths in $\mathcal{T}$ whose atomic monomials do not coincide formally with the corresponding atomic monomials in $\mathcal{S}$—this is possible the entire graph. Let then $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})'$ denote a subtree of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ which includes this set $\Lambda \subseteq \Lambda(\mathcal{T}')$, and define analogously the corresponding subtree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})' \subseteq (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. These are the subtrees which are not polynomially equivalent, and thus have different parametrisations. We define two naïve resize operators, $\mathfrak{r}_{\mathcal{T}} : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})' \mapsto (\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ and $\mathfrak{r}_{\mathcal{S}} : (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})' \mapsto (\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ which map those subtrees to the same floret. By Lemma 3.37, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})'$, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})'$ and $(\mathcal{F}, \boldsymbol{\theta}_{\mathcal{F}})$ are statistically equivalent. Note that both $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})'$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})'$ are staged trees, so a composition of these naïve resizes forms a resize $\mathfrak{r} = \mathfrak{r}_{\mathcal{S}}^{-1} \circ \mathfrak{r}_{\mathcal{T}} : (\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) \mapsto (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ between statistically equivalent staged trees. This proves 'only if'.

Now let $\mathfrak{m}$ be a transformation given by naïve swaps and resizes between two staged trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. If $\mathfrak{m}$ is a composition of swaps then Proposition 3.19 ensures polynomial equivalence and thus statistical equivalence by Lemma 3.13. If $\mathfrak{m}$ is a composition of resizes, then Lemma 3.37 yields statistical equivalence. Clearly, also for the composition of both of these operators holds that $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $\mathfrak{m}(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) = (\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ are statistically equivalent. This proves 'if'.

The claim follows. □

Clearly, a composition of the swap and resize operator as above enables us to overcome the limitations we face when using one or the other transformation exclusively: using swaps, we can change the order of events and discover which renormalisations of an underlying probability mass function are possible in a polynomial equivalence class without reparametrising the model. Using resizes, we can reparametrise and shorten redundant subtrees, or blow up a given representation to a most detailed graphical description. Resizing these subtrees might then create new twins and might enable us to swap subtrees which were spuriously fixed in the polynomial equivalence class (as in Corollary 3.31).

As a result, in the terminology from Chapter 2, Theorem 2 yields a full classification of all different tree parametrisations of the same semi-algebraic variety.

We show in the next section how the swap and resize can be applied in practice and how we can interpret the new tree-induced random variables these operators can create.

3.2.4. The Christchurch Health and Development Study (CHDS)

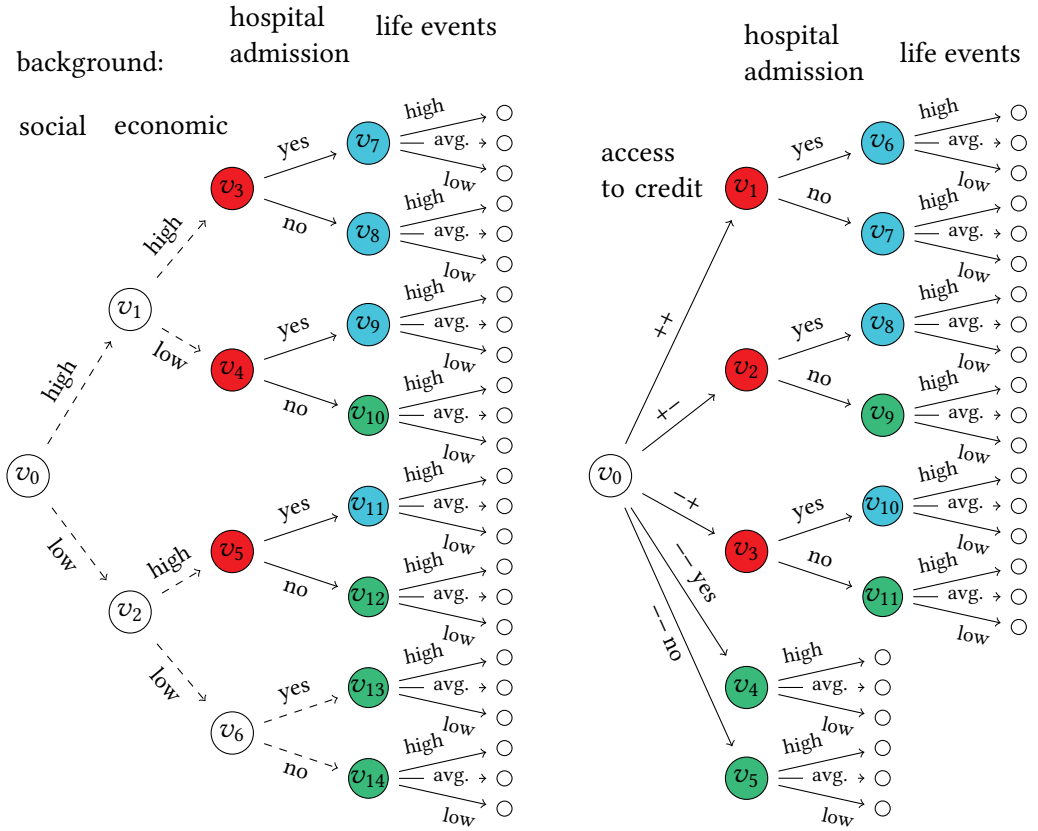
The *Christchurch Health and Development Study*—for short often the *CHDS*—followed a cohort of nearly one thousand children in Christchurch, New Zealand, over the course of thirty years, taking a whole range of measurements in order to determine the factors which drive child

illness (CHDS, 2017). The work of Fergusson et al. (1986) provides an early analysis of these data, limited to a time span of five years. We follow these authors and Barclay et al. (2013) in grouping the measurements into four very broad categories: *social background* measuring among others maternal educational level and age at childbirth, a child's ethnicity, family social class and whether a child entered an adoptive, a single or two parent family; *economic background* assessing factors such as income, standard of living, financial difficulty and the quality of accommodation; the number of stressful *life events* of a child like death, illness or unemployment in the family and marital disharmony—all of these interpreted as discrete random variables with respective states 'high', 'average' and 'low'— and the state of health of a child is assessed as *hospital admission* within that time frame 'yes' or 'no'. In Smith et al. (2017) we dedicate a full chapter to the analysis of the CHDS using staged trees and chain event graphs. The presentation in this thesis will necessarily be much shorter.

Barclay et al. (2013); Cowell and Smith (2014) provide a very nuanced analysis of the conditional independence statements which can be drawn out of the CHDS data and have shown that the staged tree representations they found represented a model which fitted the data much better than any alternative Bayesian network. In Fig. 3.7.1 we repeat this highest scoring stratified staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ which is compatible with the problem variables given above. So here, every root-to-leaf path corresponds to a possible unfolding of events in the life of a child monitored during the study.

The colouring of $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ visualises the interplay between the pre-specified problem variables. For instance, the red coloured stage $\{v_3, v_3, v_5\}$ on the second level of the tree encodes that the respective probability of hospital admission is the same for all children from a high social or economic background, whereas these probabilities change for children from a low social and economic background, $v_6 \notin \{v_3, v_3, v_5\}$. Similarly, the two vertices v_7 and v_8 in the blue stage and the two vertices v_{13} and v_{14} in the green stage encode that for children from a high social and economic background or from a low social and economic background, the number of life events does not depend on hospital admission. For children from less homogeneous backgrounds, the probabilities of a number of life events however are affected by hospital admission.

Now, $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ contains a saturated subtree with inner vertices v_0, v_1, v_2, v_6 and dashed depicted edges emanating from these. This subtree does not provide any stage information, so is in this sense superfluous to a specification of the model $\mathbb{P}_{(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})}$. Using Lemma 3.36(a) we can hence resize this subtree and obtain a different staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. This is given in Fig. 3.7.2. By Lemma 3.37, both staged trees $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ are statistically equivalent, so both are possible representations of the same highest scoring model. Note that by replacing the sat-

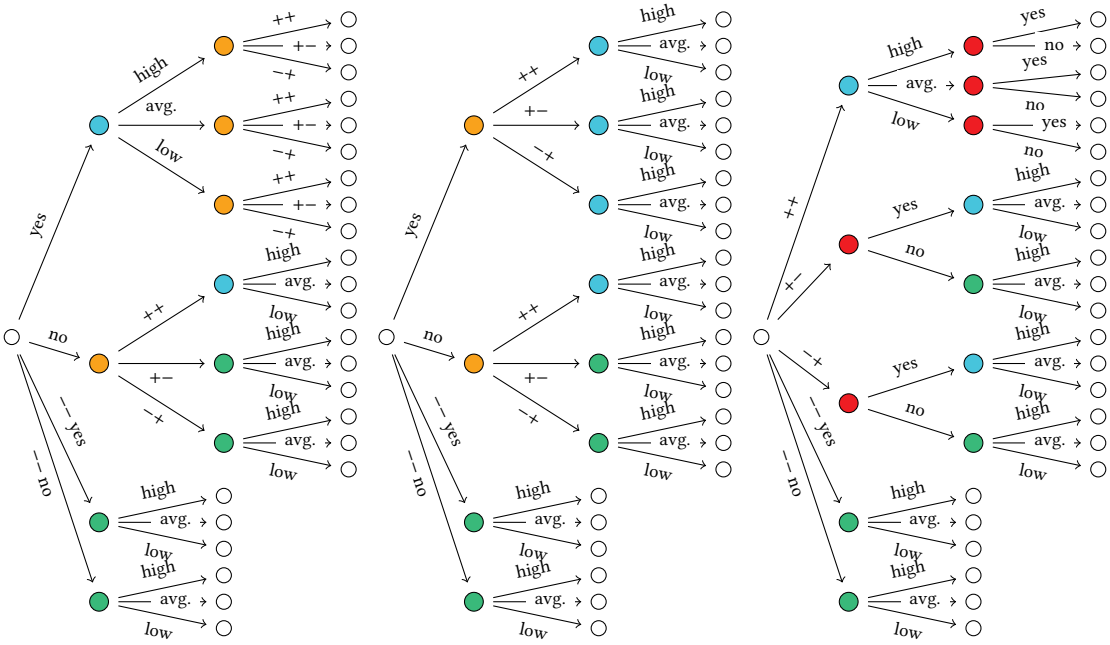


(3.7.1) The staged tree $(\mathcal{T}, \theta_{\mathcal{T}})$ found by Cowell and Smith (2014).

(3.7.2) A staged tree $(\mathcal{S}, \theta_{\mathcal{S}})$, obtained from $(\mathcal{T}, \theta_{\mathcal{T}})$ using a resize.

Figure 3.7. Two statistically equivalent staged trees for the CHDS dataset.

urated subtree by a floret, the root vertex does now not belong to the problem variable *social background* but has to be given a different interpretation. In particular, the edges $e_i = (v_0, v_i)$, $i = 1, 2, \ldots, 5$ in $(\mathcal{S}, \theta_{\mathcal{S}})$ can now be assigned a new meaning as follows: e_1, e_2 and e_3 correspond to 'social background or economic status are high' and e_4 and e_5 to 'both social background and economic status are low, hospital admission yes or no'. Hence, children passing along e_1 can be said to be 'from a wealthy background', along e_2 and e_3 'from a moderately wealthy background' and along e_4 and e_5 'from a poor background'. From the stage structure of $(\mathcal{S}, \theta_{\mathcal{S}})$ we can see that the probabilities of certain numbers of life events differ between wealthy and poor children—just as found for $(\mathcal{T}, \theta_{\mathcal{T}})$ above. Interestingly, Coudouel et al. (2002) asserts that the *access to credit* is a possible monetary measurement of poverty. This can be defined as being able to borrow from a social network or having own savings and is a natural indicator of wealth. So this assertion gives some external support from given domain



(3.8.1) $(\mathcal{S}, \theta_{\mathcal{S}})_1$ **(3.8.2)** $(\mathcal{S}, \theta_{\mathcal{S}})_2$ **(3.8.3)** $(\mathcal{S}, \theta_{\mathcal{S}})_3$

Figure 3.8. Three staged trees in the polynomial equivalence class of $(\mathcal{S}, \theta_{\mathcal{S}})$ from Fig. 3.7.2.

knowledge rather than our dataset for moving from $(\mathcal{T}, \theta_{\mathcal{T}})$ to $(\mathcal{S}, \theta_{\mathcal{S}})$, suggested from the results of the automated model search on the CHDS data.

We can then analyse the polynomial equivalence class of $(\mathcal{S}, \theta_{\mathcal{S}})$. Note first that this staged tree contains five twins: these are contained in the red coloured stage $u_{\text{red}} = \{v_1, v_2, v_3\}$, the green coloured stage $u_{\text{green}} = \{v_4, v_5\}$ and the blue coloured stage $u_{\text{blue}} = \{v_6, v_7\}$. Both of the latter two stages contain vertices other than those listed but these do not form twins. Following the results obtained in Section 3.2.1, we can now check which swaps of these twins do not violate any model assumptions and which ones are naïve operations. Clearly, any non-floret swap on u_{red} would result in a root-floret containing edges which are identified with edges elsewhere in the graph. So the resulting probability tree would then not be staged: compare also Fig. 3.3.2 of Example 3.24. We hence call $\mathfrak{s}_1$ the floret-swap which inverts the order of all three florets emanating from u_{red} and the root floret. Similarly, any swap around u_{green} would result in an identification of root-edges with edges repeated at the leaves of the tree and again the resulting labelled event tree would be neither staged nor saturated. The remaining twin u_{blue} can be swapped without violating any stage structure. We will denote the corresponding operation $\mathfrak{s}_2$.

So there are precisely two non-naïve swaps we can perform on $(\mathcal{S}, \theta_{\mathcal{S}})$. The resulting staged trees $(\mathcal{S}, \theta_{\mathcal{S}})_1 = \mathfrak{s}_2 \circ \mathfrak{s}_1(\mathcal{S}, \theta_{\mathcal{S}})$, $(\mathcal{S}, \theta_{\mathcal{S}})_2 = \mathfrak{s}_1(\mathcal{S}, \theta_{\mathcal{S}})$ and $(\mathcal{S}, \theta_{\mathcal{S}})_3 = \mathfrak{s}_2(\mathcal{S}, \theta_{\mathcal{S}})$ are depicted in Figs. 3.8.1 to 3.8.3, respectively. By Proposition 3.25, these three trees together with the original staged tree are all elements in the polynomial equivalence class of $(\mathcal{S}, \theta_{\mathcal{S}})$. So we have completely characterised $[\mathcal{S}, \theta_{\mathcal{S}}]^c$. We can now interpret these different staged trees in terms of the CHDS.

Note first that $(\mathcal{S}, \theta_{\mathcal{S}})$ and $(\mathcal{S}, \theta_{\mathcal{S}})_2$ are the only stratified representation of the model. So these are the only staged trees from which we can read problem variables level by level. In $(\mathcal{S}, \theta_{\mathcal{S}})$, these are *access to credit*, *hospital admission* and *life events*. In $(\mathcal{S}, \theta_{\mathcal{S}})_2$, we have the random variable *hospital admission (and poor)* attached to the root, then *access to credit* and finally *life events*. In $(\mathcal{S}, \theta_{\mathcal{S}})_1$ and $(\mathcal{S}, \theta_{\mathcal{S}})_3$, an interpretation of the levels of these staged trees is not as straightforward. However, we can still observe two points here. First, all staged trees in Fig. 3.8 express conditional independence statements equivalent to the ones in the staged trees in Fig. 3.7. So even though all of these representations are graphically very different, we can easily see that they indeed specify the same model—and are statistically equivalent. Second, these five different representations do not depict events in the model in just any order: in fact, there seems to be an underlying mechanism specifying that life events need to be activated *after* hospital admission. We will elaborate on this point in much more detail in Section 4.2 in the final chapter.

So the tools provided in the previous two sections, and in particular Theorem 2, enabled us to employ purely graphical means in order to traverse the statistical equivalence class of a staged tree elicited from data. We have been able to do this without having to provide numerical values for primitive or atomic probabilities, and in fact without having to resort to the original model selection algorithm in order to guarantee that our obtained results were still valid within the elicited model. This is both fast and can be done without a proper understanding of the algebra behind this approach. As a consequence, our graphical tools are of great advantage when staged trees are elicited from domain experts rather than from data, and when it would be very cumbersome to validate a large number of different staged tree representations of a problem.

In addition, we will see in the subsequent chapter that using these new methods we have been able to elicit very subtle points about the interplay of events in this setting which have not been obtainable using more standard graphical models such as Bayesian networks.

3.3. Eliciting a graph from a tree-compatible polynomial

Whilst it is most straightforward to read an interpolating polynomial—and a tree-compatible factorisation of that polynomial—from a staged tree, it is often not immediately obvious how to draw a labelled event tree when given a polynomial in distributed form. So this section will answer the question we raised on page 88: *Given a polynomial, how can we decide whether or not it is tree compatible? And how can we find—if existent—a possible tree-compatible factorisation?* If we know that a polynomial is tree compatible and if we know its nested representation then we can simply use the result in Proposition 3.19 and directly translate the given nesting of labels into a nesting of edges, or rather a labelled tree graph. So in answering the query above we will provide the mechanism for the inverse step to the theory we developed above, and will show how to infer a labelled event tree from a polynomial.

In this development we will first present a list of criteria that bound the set of all polynomials which might admit a tree-compatible factorisation. We then develop a constructive algorithm—Alg. 1 on page 115—for finding all tree-compatible factorisations of these polynomials (possibly none). These results are the focus of ongoing work, to be found in Görgen et al. (2017), and will be given here in a modified form. Using the proof of Proposition 3.19, we then state a second algorithm—Alg. 2 on page 116—for translating the tree-compatible factorisations we found above into labelled event trees.

The techniques for answering the above queries heavily rely on the notion of ideals in polynomial rings which are spanned by atomic monomials, as introduced in Section 1.3. The results we present here can therefore be translated into computational commutative algebra and allow for an implementation in freely available software: compare Section 2.3 and Chapter 2. In fact, a first version of the algorithm presented here is already available in form of the package `StatStagedTrees.cpkg5` in `CoCoA-5.1`.

We will first take the complimentary approach and, from a given labelled event tree, derive constraints which are necessary for a polynomial to be tree compatible.

So let in the following always $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a square-free labelled event tree with interpolating polynomial $c_{\mathcal{T}}$. We first observe that $c_{\mathcal{T}}$ can be recursively written as the linear sum of interpolating polynomials of subtrees of $\mathcal{T}$, so:

$$c_{\mathcal{T}}(\boldsymbol{\theta}) = \sum_{v \in \mathrm{ch}(v_0)} \theta(v_0, v) c_{\mathcal{T}(v)}(\boldsymbol{\theta}) \tag{3.40.1}$$

$$c_{\mathcal{T}(v)}(\boldsymbol{\theta}) = \sum_{v' \in \mathrm{ch}(v)} \theta(v, v') c_{\mathcal{T}(v')}(\boldsymbol{\theta}) \tag{3.40.2}$$

for all $v \in V$. This is simply because every vertex-centred event can be decomposed in the disjoint union of events centred at children of that vertex, $\Lambda(v) = \bigcup_{v' \in \mathrm{ch}(v)} \Lambda(v')$, and because of the countable additivity of the underlying probability measure—we have seen in Lemma 3.2 that both of these notions are intrinsically linked. Recursively plugging (3.40.2) into (3.40.1) yields precisely the tree-compatible factorisation (3.23) of $c_{\mathcal{T}}$ we obtained in the previous section. So for a polynomial to be tree compatible it is clearly necessary that it can be written in terms of recursive linear sums as above. So we deduce immediately from Definition 3.17 that the following is true.

Lemma 3.40 (Root expansion)**.** *Let $c \in \mathbb{R}[\Theta]$ be a tree-compatible polynomial. Then c can be written as the linear sum $c(\boldsymbol{\theta}) = \sum_{i=1}^{k} \theta_i m_i(\boldsymbol{\theta})$ of tree-compatible polynomials $m_i \in \mathbb{R}[\Theta]$ and $i = 1, \ldots, k$ where $k \geq 2$.*

In addition, every polynomial which can be written as the sum of tree-compatible polynomials as above is itself tree compatible.

The value of the almost tautological result of Lemma 3.40 lies in the fact that it gives us a clue as to where to start in the search for tree compatibility: if we can find a linear combination of a given polynomial in terms of other polynomials, and if each of these can also be written in terms of such a linear combination, then the given polynomial was tree compatible. Of course, this result alone would only allow for an exhaustive search over all labels and would need to restart from scratch every time a polynomial cannot be further decomposed as above. We present below how an ideal associated to a given polynomial can help to make this type of search much more systematic.

Before presenting this new approach, note that we can also identify a set of those polynomial which might admit a factorisation as desired. So we first list a number of constraints for polynomials to be tree compatible. These are mainly implied by the special way in which the indeterminates in the interpolating polynomial of a labelled event tree relate to their graphical representation. Compare also Remark 3.15 where we have shown that the nested bracketing in a tree-compatible factorisation mirrors precisely the nesting of florets in the corresponding tree graph. Now, for instance, every term in such a polynomial contains precisely one of the root-labels. In fact, a collection of root-to-leaf paths share a number of edges only if their monomials share the corresponding indeterminates. In addition, we can straightforwardly relate the degree of monomials to the length of a root-to-leaf path: in the notation of Definition 1.2, we say that a monomial $\boldsymbol{\theta}^{\alpha} = \theta_1^{\alpha_1}\theta_2^{\alpha_2}\cdots\theta_d^{\alpha_d}$ has *degree* k if the sum of its exponents is equal to k, so $\sum_{j=1}^{d} \alpha_j = k$. The degree of a polynomial $c = \sum_{i=1}^{n} \boldsymbol{\theta}^{\alpha_i}$ is then the maximum degree of its monomials, in symbols $\deg(c) = \max_{\alpha} \deg(\boldsymbol{\theta}^{\alpha})$. Clearly, if a root-to-leaf path has k edges, then its atomic monomial is of degree k. As a consequence, the degree of an interpolating

polynomial is the length of its longest root-to-leaf path. The number of monomials which are summed in an interpolating polynomial is the number of atoms in the underlying space. And the number of indeterminates in that polynomial is at least the number of edges in the tree—in staged trees, the identification of indeterminates with edges is of course not unique.

The proposition below uses these observations to translate a number of straightforward graphical properties of a labelled event tree into properties of its interpolating polynomial.

Proposition 3.41 (Characterising interpolating polynomials)**.** *Let* $c(\boldsymbol{\theta}) = \sum_{i=1}^{n} \boldsymbol{\theta}^{\alpha_i}$ *by a square-free polynomial,* $\boldsymbol{\theta} = (\theta_1, \dots, \theta_d)$ *and* $\alpha_i \in \{0,1\}^d$ *for all* $i = 1, \dots, n$ *where* $n, d \in \mathbb{N}$*. The following conditions are necessary for* c *to be tree compatible:*

a) $d, n \geq 2$ *and* $\deg(c) < d \leq 2n - 2$*.*

b) There are $\theta_1, \dots, \theta_k$ *with* $k \geq 2$ *such that* c *can be written as* $c = \sum_{i=1}^{k} \theta_i c_i(\boldsymbol{\theta}_{-i})$ *where* $\boldsymbol{\theta}_{-i}$ *is the vector* $\boldsymbol{\theta}$ *with entry* θ_i *deleted. In addition, every* $c_i(\boldsymbol{\theta}_{-i})$ *fulfils (a) for* $d_i = d - k$ *and* n_i *equal to the number of terms in* c_i*.*

c) For the $\theta_1, \dots, \theta_k$ *from (b) the number of terms in with which each of these indeterminates appears in the polynomial* c *is greater than the degree of the monomials in which they appear.*

d) If $\deg(\boldsymbol{\theta}^{\alpha_i}) = \deg(c)$*, then there exists* $\boldsymbol{\theta}^{\alpha_j}$ *with* $i \neq j$ *with the same degree as* $\boldsymbol{\theta}^{\alpha_i}$ *and the degree of the greatest common divisor of* $\boldsymbol{\theta}^{\alpha_j}$ *and* $\boldsymbol{\theta}^{\alpha_i}$ *is equal to* $\deg(c) - 1$*.*

e) If θ_i *and* θ_j *divide* $\boldsymbol{\theta}^{\alpha}$*, then there exists a monomial* $\boldsymbol{\theta}^{\beta}$ *in* c *which is divisible by either* θ_i *or* θ_j *but not both.*

f) No two monomials in c *can be multiples of each other.*

Proof. (a) Always, $d, n \geq 2$ because non-empty event trees have at least one floret with two labelled edges, so also two root-to-leaf paths. Then it is well-known that for any tree $\mathcal{T} = (V, E)$, the number of vertices is always one less than the number of edges, $\#E = \#V - 1$. For binary trees which have root-to-leaf paths which are all of the same length, the number of vertices is also precisely one less than twice the number of root-to-leaf paths[24], so $\#V = 2n - 1$, where $n = \#\Lambda(\mathcal{T})$. So in general, we have the inequality $\#V \leq 2n - 1$. This implies that also $d + 1 \leq \#E + 1 \leq 2n - 1$ which yields the claim.

(b) True by Lemma 3.40 because in square-free monomials the root-labels cannot be repeated within the tree.

(c) Consider Fig. 3.9. In labelled event trees, an atomic monomial of degree $l \in \mathbb{N}$ is associated to a root-to-leaf path which has $l \in \mathbb{N}$ edges. This path has one bifurcation at every vertex, so is embedded in a graph with at least $l + 1$ distinct root-to-leaf paths. So every root-label θ_1 occurs in monomials of maximal degree l and there are at least $l + 1$ of these.

24 A proof of this result is provided in the appendix: see Proposition A.3.

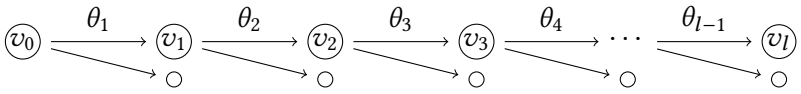


Figure 3.9. A root-to-leaf path $\lambda = (e_1, \ldots, e_l)$ in an event tree.

(d) Because $\#E(v) \geq 2$ for all $v \in V$, every leaf-floret has at least two edges. There are hence at least two monomials of the same maximal degree, namely those belonging to the longest paths in the tree: these are equal and have the same labels until they split at a leaf-floret.

(e) If c is the interpolating polynomial of a labelled event tree, then every two indeterminates θ_i and θ_j, $i \neq j$, which divide the same atomic monomial $\boldsymbol{\theta}^{\alpha}$ lie on the same root-to-leaf path. Let θ_i be a component of the vector of floret labels $\boldsymbol{\theta}_v$ and θ_j of $\boldsymbol{\theta}_w$ for vertices $v \neq w$ which lie on the same root-to-leaf path, and let v be on a lower level than w. Now in event trees $\boldsymbol{\theta}_w$ has at least two components, so there exists a second label $\theta_j' \neq \theta_j$ emanating from the same vertex w. Hence, there exists an atomic monomial $\boldsymbol{\theta}^{\beta}$ in c which is identical to $\boldsymbol{\theta}^{\alpha}$ along the subpath from the root to w and then replaces θ_j by θ_j'. This $\boldsymbol{\theta}^{\beta}$ is by construction divisible by θ_i. Because c is square-free, $\boldsymbol{\theta}^{\beta}$ is then not divisible by θ_j.

(f) Assume that there were two terms $\boldsymbol{\theta}^{\alpha}$ and $\boldsymbol{\theta}^{\beta}$ in c which are multiples of each other. Say $\boldsymbol{\theta}^{\beta} = \boldsymbol{\theta}^{\alpha}\boldsymbol{\theta}^{\gamma}$ where γ is not the null vector. Then because $\boldsymbol{\theta}^{\beta}$ is the atomic monomial associated to a root-to-leaf path, $\boldsymbol{\theta}^{\alpha}$ is either associated to a *subpath* of that root-to-leaf path (and hence not an atomic monomial) or $\boldsymbol{\theta}^{\gamma} = 1$, in which case $\gamma = 0$. This is a contradiction, so there can be no two atomic monomials which are multiples of each other. □

The conditions in Proposition 3.41 above are necessary but not sufficient. In fact, it is easy to check that there are polynomials which fulfil all criteria above but are not tree-compatible.

Example 3.42 (Proposition 3.41 is not sufficient.)**.** The polynomial

$$c = \theta_1\theta_5 + \theta_1\theta_6 + \theta_2\theta_3\theta_4 + \theta_2\theta_3\theta_6 + \theta_2\theta_4\theta_6 \tag{3.41}$$

in the ring $\mathbb{R}[\theta_1, \theta_2, \ldots, \theta_6]$ satisfies all requirements (a)–(f) in Proposition 3.41 but it cannot be written in form of a tree-compatible factorisation. It is thus not the interpolating polynomial of a labelled event tree.

So after excluding all polynomials which cannot possibly be tree compatible using the proposition above, we can now run a search over possible linear expansions of a given polynomial as in Lemma 3.40 and then recursively check if the outer factors are potential root-labels. Central to this development is the insight that the notion of a decomposition of an ideal—as defined below—is intrinsically linked to a 'decomposition' of a labelled event tree into subgraphs: so

there are tools from algebraic geometry which can be employed in a very straightforward way to obtain putative root-labels.

This development builds in particular on Theorem 4 in Section 4.7 of Cox et al. (2015). This states that every ideal $I \subseteq \mathbb{R}[\Theta]$ can be written as a finite intersection of ideals $I = \bigcap_{k=1}^{m} J_k$ such that each ideal J_k in this intersection has the property that if a product of polynomials $f \cdot g \in J_k$ is contained in that ideal then either one of the factors was an element $f \in J_k$ or a multiple of one of the factors was an element of the ideal, $g^l \in J_k$ for some power $l \in \mathbb{N}$. The J_k, $k = 1, \ldots, m$, are called *primary ideals* and the intersection $I = \bigcap_{k=1}^{m} J_k$ is thus called a *primary decomposition* of the original ideal. In very simple terms, the notion of a primary ideal is a great generalisation of a prime number, and determining a primary decomposition of an ideal is analogous to writing a number as the product of prime numbers.

Now by Remark 3.15 and Proposition 3.41, edge labels in an interpolating polynomial—and especially common divisors of terms in such a polynomial—mirror the nesting of edges with these labels in a corresponding labelled event tree. We show now how this nesting can be translated into the nesting of ideals where lower levels are inferred from upper levels using the decomposition above.

We henceforth denote by $I_c = \langle \boldsymbol{\theta}^{\alpha_i} \mid i = 1, \ldots, n \rangle \subseteq \mathbb{R}[\Theta]$ the ideal spanned by all atomic monomials in a given interpolating polynomial $c = \sum_{i=1}^{n} \boldsymbol{\theta}^{\alpha_i}$ in the ring $\mathbb{R}[\Theta] = \mathbb{R}[\theta_1, \ldots, \theta_d]$. So all elements in this ideal are polynomial combinations of these monomials, and in particular the interpolating polynomial itself is an element of that ideal, $c \in I_c$. We can show that the following is true:

Proposition 3.43 (Decomposition)**.** *Let $c = c_{\mathcal{T}}$ be the interpolating polynomial of a square-free staged tree. Then one of the primary ideals in the primary decomposition of I_c is generated by the root-labels of the corresponding tree.*

It is a very rare property of an ideal to contain in its primary decomposition an ideal which is generated by degree-one indeterminates. That this is the case for the decomposition of an ideal arising from a staged tree is entirely due to the very special form of tree-compatible polynomials and because our ideals are always generated by square-free monomials[25]. We provide a proof of this rather technical result in Görgen et al. (2017).

In this thesis we will limit ourselves to presenting the general idea in an example below.

25 The assumption of square-freeness is especially important in our implementation of this result because the computation of the primary decomposition of an ideal as implemented in `CoCoA` relies on the notion of the *Alexander Dual* of an ideal: this is defined only for ideals generated by square-free monomials (Miller and Sturmfels, 2005).

Example 3.44 (Primary decomposition)**.** In Example 3.24 we analysed three different tree-compatible factorisations of the polynomial

$$c = \theta_0 + \theta_1\phi_1 + \theta_1\phi_2 + \theta_2\phi_1 + \theta_2\phi_2 + \theta_3\phi_1 + \theta_3\phi_2. \tag{3.42}$$

The atomic monomials in this polynomial induce an ideal

$$I_c = \langle \theta_0, \theta_1\phi_1, \theta_1\phi_2, \theta_2\phi_1, \theta_2\phi_2, \theta_3\phi_1, \theta_3\phi_2 \rangle \tag{3.43}$$

in the ring $\mathbb{R}[\theta_0, \theta_1, \theta_2, \theta_3, \phi_1, \phi_2]$ whose indeterminates are the edge labels of potential tree representations. We can now use `CoCoA` to determine the primary decomposition of the ideal above. In fact, the command

```
Use R ::= QQ[t[0..3],f[1..2]];
Ic := Ideal(t[0],t[1]*f[1],t[1]*f[2],t[2]*f[1],t[2]*f[2],t[3]*f[1],
            t[3]*f[2]);
PrimaryDecomposition(Ic);
```

yields that the ideal can be written as the intersection

$$I_c = \langle \theta_0, \theta_1, \theta_2, \theta_3 \rangle \cap \langle \theta_0, \phi_1, \phi_2 \rangle \tag{3.44}$$

of primary ideals.

So every polynomial $f \in I_c$ in our ideal is both of the form $f = \theta_0 f_0 + \theta_1 f_1 + \theta_2 f_2$ and of the form $f = \phi_1 g_1 + \phi_2 g_2$ for fitting polynomials f_0, f_1, f_2, g_1, g_2 in the ring above. These polynomial combinations are linear combinations as in Lemma 3.40: so if we wanted to use the primary decomposition above in order to test whether or not c was tree compatible, we would now need to check for $f = c \in I_c$ whether f_0, f_1, f_2 or g_1, g_2 were tree compatible. In this example, these polynomials are simple linear sums of indeterminates, so are trivially tree compatible. In general, we can check for tree compatibility for instance using recursive decompositions of these polynomials which eventually yield a tree-compatible factorisation of c: see below.

By Proposition 3.43, if c was a tree-compatible polynomial then one ideal in the primary decomposition (3.43) must be generated by the root-labels of a corresponding tree representation. So (3.44) yields two putative sets of root-labels: $\{\theta_0, \theta_1, \theta_2, \theta_3\}$ or $\{\theta_0, \phi_1, \phi_2\}$. By the results in Example 3.24, these are indeed root labels of possible tree representations of tree-compatible factorisations of the polynomial c: namely those given in Figs. 3.3.1 and 3.3.3.

Note that the decomposition above automatically excluded the root-labels $\{\theta_0, \theta_1, \phi_1, \phi_2\}$ of the labelled event tree in Fig. 3.3.2 which was not staged. This exclusion of 'repetitive' labels is

Algorithm 1: Inferring a tree-compatible factorisation from a given polynomial.

Input : A polynomial $c = \sum_{i=1}^{k} \boldsymbol{\theta}^{\alpha_i}$.
Output: A staged tree-compatible factorisation $s(c(\boldsymbol{\theta}))$ as in (3.25), if existent.

```
if deg(c) = 1 then
    s(c) = c is a linear sum of indeterminates.
else
    while deg(c) > 1 do
        determine a primary decomposition of I_c
        pick a primary ideal of the form ⟨θ_1, ..., θ_k⟩ from that decomposition
        set a vector of putative root labels to be (θ_1, ..., θ_k)
        for all vectors of putative root labels do
            write c = Σ_{i=1}^{k} θ_i c_i
            for i = 1, ..., k do
                set c_i = c
                return to 1
check whether the obtained nesting fulfils the assumptions of Corollary 3.20
```

systematic but does not guarantee to yield a labelled event tree which is always either saturated or staged. In Görgen et al. (2017) we provide a much more technical presentation and a proof of this result.

Algorithm 1 is written in pseudo-code to formalise the ideas in the example above for implementation. The algorithm is given here in a very simplified form. An implementation of this algorithm is now available in the `CoCoA` package `StatStagedTrees.cpkg5`.

Our main idea is to use primary decomposition of an ideal to find putative vectors of root labels. For each of these vectors, we then project onto one label and reiterate the procedure for the resulting polynomial. In terms of the tree graph, we hereby discover a putative root floret first and subsequently move along its emanating edges, repeating the initial step again for each vertex which is a child of that root, until we reach a leaf. If this process fails at any step, then the chosen putative root labels were not root labels and we need to repeat the above using a different candidate set of labels. If the algorithm fails for all candidate sets, then the input polynomial was not tree compatible. By construction, this algorithm always terminates because the input polynomial is of finite degree and each iteration reduces the degree by one.

This procedure is much faster than an exhaustive search which uses a generic tree-compatible factorisation and therein permutes the assignment of labels until a multiplied form of that polynomial coincides with the input polynomial. In such a search, the number of subsets of a set of labels—so the number of all sets which might be labels of a root-floret—is of order 2^d where

Algorithm 2: Building a labelled event tree from a polynomial given in tree-compatible factorisation.

Input : A tree-compatible factorisation $s(c(\boldsymbol{\theta}))$ of a polynomial as in (3.25) obtained from Alg. 1.

Output: An event tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}}) = \mathfrak{c}(s(c(\boldsymbol{\theta}))$ with labels $\boldsymbol{\theta}_{\mathcal{T}} = \boldsymbol{\theta}$ and interpolating polynomial $c = c_{\mathcal{T}}$.

```
draw a floret which has one labelled edge for every indeterminate in the set A_0 in s(c(θ))
for every index set A_j(θ) in s(c(θ)) do
    if there is only one subsum in s(c(θ)) over these indices then
        draw a floret F_j which has one labelled edge for every indeterminate in A(θ_j)
    else
        draw identical copies of the floret F_j as in 4 above, in number equal to the number
          of times the index set A_j(θ) appears in s(c(θ))
for every subsum θ_j · Σ_{θ_{j+1} ∈ A(θ_j)} θ_{j+1} in (3.25) do
    connect the edge labelled θ_j to the root of a floret F_j
```

$d \in \mathbb{N}$ denotes the number of indeterminates in the polynomial. This is much larger than the number of candidate sets provided by primary decomposition.

So for a given tree-compatible polynomial, Alg. 1 yields all tree-compatible factorisations which can be mapped to staged trees. We can now provide these factorisations as an input to Alg. 2 in order to obtain the corresponding labelled event trees. This result is based entirely on the proof of Proposition 3.19 and is repeated here for completeness.

We conclude this section by providing an illustration for how Alg. 1 works on a bigger example. In particular, we can now use the above results in order to draw out the same staged trees we have found in Section 3.2.4 when using the swap operator rather than ideal decomposition to determine all elements of a given polynomial equivalence class.

Example 3.45 (A polynomial equivalence class for the CHDS)**.** Consider again the staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ in Fig. 3.7.2, representing the CHDS study from Section 3.2.4. Its interpolating polynomial is the sum of all atomic monomials

$$\begin{aligned} c_{\mathcal{S}}(\boldsymbol{a}, \boldsymbol{h}, \boldsymbol{l}) \;=\; & a_1h_1l_1 + a_1h_1l_2 + a_1h_1l_3 + a_1h_2l_1 + a_1h_2l_2 + a_1h_2l_3 \\ & + a_2h_1l_1 + a_2h_1l_2 + a_2h_1l_3 + a_2h_2l_4 + a_2h_2l_5 + a_2h_2l_6 \\ & + a_3h_1l_1 + a_3h_1l_2 + a_3h_1l_3 + a_3h_2l_4 + a_3h_2l_5 + a_3h_2l_6 \\ & + a_4l_4 + a_4l_5 + a_4l_6 + a_5l_4 + a_5l_5 + a_5l_6 \end{aligned} \tag{3.45}$$

where $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5)$, $\boldsymbol{h} = (h_1, h_2, h_3, h_4)$ and $\boldsymbol{l} = (l_1, l_2, l_3, l_4, l_5, l_6)$ denote the respective (conditional) probabilities of different degress of **a**ccess to credit, **h**ospital admission and numbers of **l**ife events. The numbering above arises from reading the primitive probabilities (and atomic monomials) from top to bottom and left to right as depicted in Fig. 3.7.2.

We can now provide the above polynomial as an input to Alg. 1. This yields four different tree-compatible factorisations:

$$\begin{aligned} s_0(c_{\mathcal{S}}) &= a_1(h_1(l_1 + l_2 + l_3) + h_2(l_1 + l_2 + l_3)) \\ &\quad + a_2(h_1(l_1 + l_2 + l_3) + h_2(l_4 + l_5 + l_6)) \\ &\quad + a_3(h_1(l_1 + l_2 + l_3) + h_2(l_4 + l_5 + l_6)) \\ &\quad + a_4(l_4 + l_5 + l_6) + a_5(l_4 + l_5 + l_6) \end{aligned} \tag{3.46.1}$$

$$\begin{aligned} s_1(c_{\mathcal{S}}) &= h_1(l_1(a_1 + a_2 + a_3) + l_2(a_1 + a_2 + a_3) + l_3(a_1 + a_2 + a_3)) \\ &\quad + h_2(a_1(l_1 + l_2 + l_3) + a_2(l_3 + l_4 + l_5) + a_3(l_3 + l_4 + l_5)) \\ &\quad + a_4(l_4 + l_5 + l_6) + a_5(l_4 + l_5 + l_6) \end{aligned} \tag{3.46.2}$$

$$\begin{aligned} s_2(c_{\mathcal{S}}) &= h_1(a_1(l_1 + l_2 + l_3) + a_2(l_1 + l_2 + l_3) + a_3(l_1 + l_2 + l_3)) \\ &\quad + h_2(a_1(l_1 + l_2 + l_3) + a_2(l_3 + l_4 + l_5) + a_3(l_3 + l_4 + l_5)) \\ &\quad + a_4(l_4 + l_5 + l_6) + a_5(l_4 + l_5 + l_6) \end{aligned} \tag{3.46.3}$$

$$\begin{aligned} s_3(c_{\mathcal{S}}) &= a_1(l_1(h_1 + h_2) + l_2(h_1 + h_2) + l_3(h_1 + h_2)) \\ &\quad + a_2(h_1(l_1 + l_2 + l_3) + h_2(l_4 + l_5 + l_6)) \\ &\quad + a_3(h_1(l_1 + l_2 + l_3) + h_2(l_4 + l_5 + l_6)) \\ &\quad + a_4(l_4 + l_5 + l_6) + a_5(l_4 + l_5 + l_6). \end{aligned} \tag{3.46.4}$$

Let then again $\mathfrak{c}$ denote the map from Proposition 3.19 which identifies a tree-compatible factorisation with a staged tree. Then using Alg. 2, we obtain that the factorisations above correspond precisely to the staged trees $\mathfrak{c}(s_0(c_{\mathcal{S}})) = (\mathcal{S}, \theta_{\mathcal{S}})$, $\mathfrak{c}(s_1(c_{\mathcal{S}}) = (\mathcal{S}, \theta_{\mathcal{S}})_1$, $\mathfrak{c}(s_2(c_{\mathcal{S}})) = (\mathcal{S}, \theta_{\mathcal{S}})_2$ and $\mathfrak{c}(s_3(c_{\mathcal{S}})) = (\mathcal{S}, \theta_{\mathcal{S}})_3$ depicted in Fig. 3.8. So Algs. 1 and 2 provide us with the exact same set of polynomially equivalent staged trees as the analysis in Section 3.2.4.

We can see above that our primary decomposition-based algorithm is the computational counterpart to the graphical operation of using the swap operator. A direct implementation of the algebra underlying the swap operator as developed in Section 3.2.1 has so far not been successful due to the technicalities involved in dealing with non-commutative nested factorisations in a commutative framework: see also a number of comments made in the relevant section above.

At the time of writing, an extension of Alg. 1 which checks within a tree-compatible factorisation for subtrees fulfilling the requirements of Lemma 3.36—conditions for applying non-naïve resizes—is under development. This additional tool will enable us to implement the resize operation in the algorithm above and thus traverse the whole statistical equivalence class of a staged tree using ideal decomposition and projection operations.

4. Causal inference in staged tree models

In this chapter, we embed our newly developed understanding of differential and algebraic mechanisms in staged tree models into a causal framework.

Over the last thirty years there have been great advances in establishing and applying a sound mathematical framework for the rather philosophical notion of *causality*, or causal effects between a given set of problem variables. The main question scientists posed was how a change in a causal variable influences changes in the effect variables, where effect is measured relative to an observed system. Three main areas of research are involved in developing statistical methodology for this type of analysis: *counterfactuals*, *structural equation models* and *graphical models* (Pearl, 2009).

For Spirtes et al. (1993), cause and effect are a transitive, irreflexive and antisymmetric relation between random variables, and this relation can be depicted in a certain directed graph. Within this graph, an edge or a sequence of edges which connect two variables (depicted as vertices) can be interpreted as one variable directly or indirectly influencing another. Similarly, Pearl (1995a) introduces causal statistical models described by a set of equations which can be depicted by a graph having a directed edge from one variable to another whenever the calculation of quantities depending on the second requires information about the values of the first. Both of these approaches provide a functional way of expressing relationships within the class of graphical models. They also provide a framework in which an application of the *do-operator*, as defined below, formally hypothesises the impact on the whole system of controlling certain subsets of the problem variables to take certain values. This impact is relative to the case in which the observed system is not subject to any control—it is *idle*. Under the hypothesis that certain properties of the idle system remain invariant in the controlled system, the modeller is then able to both infer cause and effect and to analyse counterfactual statements of the type 'what would have happened if…' from the underlying equations (Pearl, 2000).

These formalisations did not simply provide mathematical expressions of a causal model. They also provided a formal tool to determine when it was plausible to hypothesise the potential effects of applying various treatments to a population which was only partially observed and to capture the consequent extent of the effect of the control as a function of what was observed. In practice, results like the *back-door criterion* (Pearl, 1993) established conditions

under which a probability depending on the do-operator can be calculated from conditional probabilities, and hence be estimated from data. For such an identified causal effect, a randomised trial would hence be unnecessary: this is a great advantage for instance when the design of such a trial would be very costly or unethical[26]. When it is possible to perform this type of calculation we say that a system is *causally identifiable*. Causal inference and causal identifiability continue to be very active areas of research: see for instance Silva and Evans (2014) for bounds on the average causal effect of one variable on another, Constantinou and Dawid (2015) for a link between generalised conditional independence and causal inference in a decision-theoretic framework, or Maathuis and Colombo (2015) for a recent generalisation of the back-door theorem to Markov-equivalence classes of acyclic digraphs. So at least for the standard classes of graphical models the mathematics that enable us to analyse causal structures have at last been fairly fully formalised and numerously applied, mainly in medical statistics and economics (Berzuini et al., 2012).

In this final chapter, we initialise a development that will allow us to apply the methodologies obtained above to causal inference in the more general class of staged tree models. In particular, in Section 4.1 we will use the differential approach for performing causal interventions analogous to the do-operator on a staged tree using the interpolating polynomial, and in Section 4.2 we will draw putative causal hypotheses out of the statistical equivalence class analysed in Section 3.2.4.

4.1. Causal interventions on staged trees

In the modern day a causal formalism based on probability trees was first attempted by Shafer (1996). Inspired by this seminal work, a framework for performing causal manipulations in staged trees has been developed by Thwaites et al. (2010) and has then been formalised stating causal identifiability criteria by Thwaites (2013). We repeat the most basic points in these developments below and then enhance the current literature by linking these results to the differential approach presented in Section 3.1. To our knowledge, differential methods have not been applied to causal manipulation operations before and vice versa the differential framework of Darwiche (2003) has not before been applied to causal inference. The methodologies presented here have been published in a much more condensed form in Görgen and Smith (2016).

26 Pearl presents an illustrative example around the question whether smoking 'causes' lung cancer (Pearl, 2000). Clearly if this was the case then within a randomised trial we would not want to force people to smoke—and thereby make them ill—in order to be able to observe their cancer rate.

Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree with graph $\mathcal{T} = (V, E)$ representing a population before any intervention takes place. Following Pearl (2000), we call the model represented by that tree the *idle* system. The *manipulated* staged tree is then a labelled event tree with the same graph as the idle staged tree which inherits all edge-labels that are not affected by the manipulation we propose below. Most often, manipulations of staged trees are centred on an edge such that after invention all units which in the idle system have arrived at the tail-end of that edge are now forced to pass along it and to not follow any alternative unfoldings from that vertex.

The causal hypothesis then simply asserts that the subtree of the staged tree describing the future development of that unit at that vertex is the same as it would be were that unit to have arrived in that situation naturally, but that the earlier development of the unit before it reaches the vertex will remain unchanged by the control. Explicitly, following Thwaites (2013), we will work with the following semantics in the notation of Chapter 1.

Definition 4.1 (Causal intervention and causal effect)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be an idle probability tree with graph $\mathcal{T} = (V, E)$ and let $\hat{e} = (\hat{v}, \hat{v}') \in E$ be an edge. An *intervention* on the edge $\hat{e}$ results in a *manipulated* labelled event tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\hat{e}}$ with the same graph $\mathcal{T}_{\hat{e}} = \mathcal{T}$ where now all edges emanating from $\hat{v}$ are assigned probability zero, $\theta(e) = 0$ for all $e \in E(\hat{v}) \setminus \{\hat{e}\}$, except for the manipulated edge which is assigned probability one, $\theta(\hat{e}) = 1$. All other edge labels in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\hat{e}}$ are inherited from $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$.

The *causal effect* of this intervention on an event $A \subseteq \Lambda(\mathcal{T})$ is the probability of that event A under the new probability measure $\Pi_{\boldsymbol{\theta}, \mathcal{T}}(A \,||\, \Lambda(\hat{e}))$ associated to the manipulated tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\hat{e}}$.

By definition, after intervention we obtain a 'degenerate' probability tree which has edge labels that are equal to zero or one. We show in Smith et al. (2017) that such a manipulated tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{\hat{e}}$ is statistically equivalent to a (non-degenerate) probability tree denoted $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})_{-\hat{e}}$ where the manipulated floret and all root-to-leaf paths which have been assigned probability zero have been cancelled and the manipulated edge $\hat{e} = (\hat{v}, \hat{v}')$ has been contracted such that $\hat{v} = \hat{v}'$.

As is common in causal literature, we use the symbol || rather than | to distinguish an enforced control from conditioning on an observation (Pearl, 2000). These are in fact two very different operations: the latter requires for instance that the probability labels in the subtree be renormalised, so divided by the probability of arriving at that vertex as calculated in Section 3.1.

Definition 4.1 is set up to contain the notion of causal intervention and effect originally proposed by Pearl (1995b) as a special case. For Pearl, causal manipulations on an acylic digraph representing a Bayesian network model are always (compositions of) *atomic controls* denoted

$\text{do}(X_i = \hat{x_i})$ where the *do-operator* expresses the control that a random variable X_i is externally forced to take a certain value $\hat{x_i} \in \mathbb{X}_i$ for one $i = 1, \ldots, m$ in the notation from Section 1.2.2. The probability mass function p on a vector of random variables $X = (X_1, \ldots, X_m)$ after intervention then changes to be

$$p(x_1, \ldots, x_m \mid\mid \hat{x_i}) = \begin{cases} \dfrac{p(x_1, \ldots, x_i, \ldots, x_m)}{p(x_i \mid x_{\text{pa}(i)})} & \text{if } x_i = \hat{x_i} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

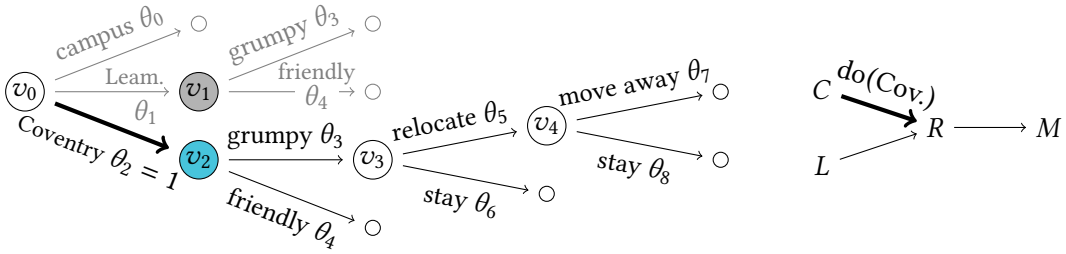
for $(x_1, \ldots, x_m) \in \mathbb{X}$. According to Pearl, this manipulation is equivalent to manually removing the link between the random variable X_i and its parent vertices $X_{\text{pa}(i)}$ in the acyclic digraph with vertices X whilst keeping the rest of the network intact. In this way, only the descendants of X_i can be affected by this manipulation. Just as in Definition 4.1, the causal hypothesis here simply asserts that the development of the descendants of X_i in the graph is the same as it would have been had that variable taken the value $\hat{x_i}$ naturally.

There are many circumstances where Pearl's atomic control is not sufficiently general. For instance, a scientist might want to assign a certain treatment $X_i = \hat{x_i}$ exclusively to patients with a particular history $X_A = x_A$ for some index set $A \subseteq \text{pa}(i)$. This would require a conditional or 'context-specific manipulation' $\text{do}(X_{i, X_A = x_A} = \hat{x_i})$ where $X_{i, X_A = x_A}$ denotes the conditional random variable $X_i | X_A = x_A$. This control cannot be straightforwardly expressed using (4.1). In fact, in an X-compatible staged tree representation of a Bayesian network, the standard intervention $\text{do}(X_i = \hat{x_i})$ would correspond to a simultaneous intervention on all vertices associated with X_i, forcing every unit in the system represented by that tree along an edge labelled $\hat{x_i}$. This is an intervention on all vertices along the same level of a stratified tree which would force the atomic control $X_i = \hat{x_i}$ independent of the context.

In contrast, a tree intervention as proposed in Definition 4.1 above is far more flexible. By choosing one particular situation, we are now able to execute the manipulation $X_i = \hat{x_i}$ precisely for patients with the history $X_A = x_A$, enforcing only the edge labelled $\hat{x_i}$ which is located in a subtree rooted after the unfolding $X_A = x_A$. Although contingent manipulations have subsequently been studied in Bayesian networks, causal manipulation can be expressed much more simply within staged tree representations. In particular, because staged trees do not rely on a set of a priori problem variables, manipulations can also be analysed in terms of any event of interest.

Consider a toy example we have developed to illustrate this point.

Example 4.2 (Causal manipulations)**.** We assume that first year students at Warwick University can find accommodation either on campus, in Coventry or in Leamington Spa. Landlords



(4.1.1) An idle staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. After controlling the edge $\hat{e} = (v_0, v_2)$, the manipulated tree includes labels zero and one.

(4.1.2) A do-operation in an acyclic digraph.

Figure 4.1. Two graphical representations for the causal manipulation in Example 4.2.

in both cities can be either friendly or grumpy. If they are grumpy then students might consider moving house, and if they do move then they might also consider leaving the city they live in. We shall assume that the attitude of landlords is the same in both cities, such that the probability of renting with a friendly landlord does not depend on the location. We also assume that we are only interested in the flow of students in Coventry. We can represent this setting by the idle staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 4.1.1 where the vertices v_1 and v_2 are in the same stage because their attached transition probabilities are equal by assumption.

Alternatively, we can model this setting using an acyclic digraph as in Fig. 4.1.2 whose vertices correspond to the random variables *city* C taking values 'Coventry' and 'Leamington Spa', *landlord* L taking values 'grumpy' and 'friendly', *relocating* within the city R and *moving* away M, both taking values 'yes' and 'no'. Just like in Example 1.12, this graph obscures the asymmetries inherent to the problem.

Suppose we are now interested in what would happen were students forced to live in Coventry, for instance by a university policy which aims to cut commuting times. Using a staged tree model, under this policy the edge $\hat{e} = (v_0, v_2)$ in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ would depict the only possible unfolding from the root. This controlled edge will be assigned probability one. The alternative unfoldings emanating from other edges of the root $E(v_0) \setminus \{\hat{e}\}$ will be assigned probability zero—so the all atoms represented by root-to-leaf paths going through these edges will have probability zero as well. These now impossible unfoldings have been greyed out in Fig. 4.1.1. When contracting the edge (v_0, v_2) that every unit has to pass along to one vertex $v_0 = v_2$, the manipulated system is then given by the induced subtree $(\mathcal{T}(v_2), \boldsymbol{\theta}_{\mathcal{T}(v_2)})$ rooted at v_2 as part of the idle system.

In the acyclic digraph representation, the same policy would enforce the city variable C to take the value Coventry. So this intervention is an atomic control do(C = Coventry). Note that from Fig. 4.1.2 it is not immediately clear which unfoldings will be assigned probability zero in the controlled system. We can only deduce that the variable landlord L will not be affected.

Within the Bayesian network, the effect of this intervention can now be calculated using the semantics of the do-operator developed by Pearl (2000). In the staged tree however the effect can be determined in a much more straightforward way and can in fact immediately be read from the graph. So here when controlling (v_0, v_2), atomic probabilities depicted in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ are projected onto the subtree $(\mathcal{T}(v_2), \boldsymbol{\theta}_{\mathcal{T}(v_2)})$. Hence,

$$\begin{aligned}\boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}} &= (\theta_0, \theta_1\theta_3, \theta_1\theta_4, \theta_2\theta_3\theta_5\theta_7, \theta_2\theta_3\theta_5\theta_8, \theta_2\theta_3\theta_6, \theta_2\theta_4) \\ &\mapsto (0, 0, 0, \theta_3\theta_5\theta_7, \theta_3\theta_5\theta_8, \theta_3\theta_6, \theta_4) = \boldsymbol{\pi}_{\boldsymbol{\theta},\mathcal{T}(v_2)}\end{aligned} \tag{4.2}$$

is the new probability mass function $\pi_{\boldsymbol{\theta},\mathcal{T}}(\,\cdot\,||\Lambda(\hat{e}))$ after intervention, as in Definition 4.1.

Using the differential framework and the notation set up in Section 3.1, we can now express the vertex-intervention from Definition 4.1 in terms of a differential operation on the interpolating polynomial. So effects of causal interventions can easily be calculated using simple polynomial operations.

Lemma 4.3 (Differential manipulations)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a staged tree, $\mathcal{T} = (V, E)$, and $c_{\mathcal{T}}$ its interpolating polynomial. Suppose we enforce a control on an edge $\hat{e} \in E$. Then the causal effect on any event $\iota_{\mathcal{T}}(A) \subseteq \Lambda(\mathcal{T})$ can be calculated as the differential operation*

$$P_{\boldsymbol{\theta}}(A \,||\, A(\hat{e})) = \frac{\partial^2 c_{\mathbb{1}_{\iota_{\mathcal{T}}(A)},\mathcal{T}}(\boldsymbol{\theta}, \boldsymbol{\epsilon})}{\partial\theta(\hat{e})\partial\epsilon(\hat{e})} \tag{4.3}$$

where $A(\hat{e})$ denotes the event represented by $\Lambda(\hat{e})$.

The proof of this result follows the same rationale as the results presented in Section 3.1, this time translating the manipulation in Definition 4.1 into a differential operation. In a sense, this operation is much easier than conditioning: we simply take the derivative with respect to the label of the manipulated edge. This sets that label to one and automatically cancels out all complimentary events while leaving the remaining labels invariant.

Example 4.4 (Example 4.2 continued.)**.** The interpolating polynomials for the idle $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ and the manipulated staged tree $(\mathcal{T}(v_2), \boldsymbol{\theta}_{\mathcal{T}(v_2)})$ in Fig. 4.1.1 equal

$$\begin{aligned}c_{\mathcal{T}}(\boldsymbol{\theta}) &= \theta_0 + \theta_1\theta_3 + \theta_1\theta_4 + \theta_2\theta_3\theta_5\theta_7 + \theta_2\theta_3\theta_5\theta_8 + \theta_2\theta_3\theta_6 + \theta_2\theta_4, \\ \frac{\partial c_{\mathcal{T}}(\boldsymbol{\theta})}{\partial\theta_2} &= \theta_3\theta_5\theta_7 + \theta_3\theta_5\theta_8 + \theta_3\theta_6 + \theta_4 = c_{\mathcal{T}(v_2)}(\boldsymbol{\theta}),\end{aligned} \tag{4.4}$$

respectively, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_8)$. If we are now interested in the probability of a student leaving their accommodation, assuming she was initially forced to live in Coventry, we can

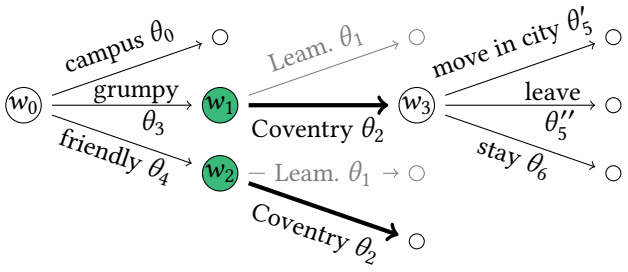


Figure 4.2. A staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ which is statistically equivalent to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ from Fig. 4.1.1.

calculate that

$$\Pi_{\boldsymbol{\theta},\mathcal{T}}(\Lambda(v_4) \mid\mid \Lambda(\hat{e})) = \frac{\partial^2 c_{\mathcal{T}(v_4)}(\boldsymbol{\theta})}{\partial\theta_2\partial\epsilon(\hat{e})} = \theta_3\theta_5 \tag{4.5}$$

where v_4 is the vertex associated with the event 'relocating' and $\hat{e} = (v_0, v_2)$ denotes again the manipulated edge. So this probability equals exactly the product of edge labels on the subpath from v_2 to v_4. It is thus simply the probability of passing from the root of the manipulated system to the vertex depicting the event of interest.

In the example above, edge manipulations in staged trees translate straightforwardly into subgraphs. To illustrate that this is not generally the case, suppose that we happen to make an unwise but statistically equivalent choice of representing our model where graphical manipulations are a lot less straightforward.

Example 4.5 (Example 4.4 continued.)**.** The staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ in Fig. 4.2 is statistically equivalent to $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ from Fig. 4.1.1 via a swap and a resize operation as in Section 3.2:

$$\begin{aligned} c_{\mathcal{T}}(\boldsymbol{\theta}) &= \theta_0 + \theta_1(\theta_3 + \theta_4) + \theta_2(\theta_3(\theta_5(\theta_7 + \theta_8) + \theta_6) + \theta_4) \\ &= \theta_0 + \theta_3(\theta_1 + \theta_2(\theta_5(\theta_7 + \theta_8) + \theta_6)) + \theta_4(\theta_1 + \theta_2) \\ &= \theta_0 + \theta_3(\theta_1 + \theta_2(\theta_5' + \theta_5'') + \theta_6)) + \theta_4(\theta_1 + \theta_2) = c_{\mathcal{S}}(\boldsymbol{\theta}') \end{aligned} \tag{4.6}$$

where $\boldsymbol{\theta} = (\theta_0, \theta_1, \ldots, \theta_8)$, $\boldsymbol{\theta}' = (\theta_0, \theta_1, \ldots, \theta_4, \theta_5', \theta_5'', \theta_6)$ and $\theta_5' = \theta_5\theta_7$, $\theta_5'' = \theta_5\theta_8$.

In this alternative representation $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$, the causal manipulation of forcing a student to live in Coventry is not a simple edge intervention as in Definition 4.1 and cannot be depicted as a subtree of $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$. This manipulation is in fact a composite intervention on the thick depicted edges in Fig. 4.2 which are not connected. It is easy to check that a partial derivative with respect to θ_2 however still yields the same result as above: $\frac{\partial}{\partial\theta_2} c_{\mathcal{S}}(\boldsymbol{\theta}') = \theta_3\theta_5$ just like in (4.5). So when dropping the $\boldsymbol{\epsilon}$-indicators in the interpolating polynomial, the result of Lemma 4.3 can be used to express multiple interventions on edges with the same label: we formalise this observation in Proposition 4.7 below.

Similarly, or worse, if for instance in the framework of a mental health campaign we were interested in assessing the effect of forcing students who are renting with a grumpy landlord to move house, then this could be easily expressed in manipulating the edge (v_3, v_4) in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$. However, again $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ would not allow us to answer this query because this time we would force a unit to go through two mutually exclusive edges simultaneously, following two different unfoldings from w_3. This operation would not be meaningful in any context—it is not 'valid' in the terminology we introduce below.

In Görgen and Smith (2016) we proposed a new local operation which combines the development of Section 3.2 where we replaced labelled tree graphs by nested polynomial representations with the differential semantics for edge-interventions discussed above.

Definition 4.6 (Local manipulation)**.** Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a probability tree with graph $\mathcal{T} = (V, E)$ and an edge $\hat{e} = (\hat{v}_{j-1}, \hat{v}_j) \in E$, and let $c_{\mathcal{T}}$ be its interpolating polynomial. We denote by

$$\mathfrak{d}(\hat{e}, c_{\mathcal{T}}) = \sum_{(v_0, v_1) \in E(v_0)} \theta(v_0, v_1) \left(\cdots \left(\frac{\partial}{\partial \theta(\hat{e})} \sum_{(v_{j-1}, v_j) \in E(v_j)} \theta(v_{j-1}, v_j) \left(\cdots \left(\sum_{(v_{k-1}, v_k) \in E(v_{k-1})} \theta(v_{k-1}, v_k) \right) \right) \right) \right) \tag{4.7}$$

a local differentiation on a tree-compatible factorisation of $c_{\mathcal{T}}$ which is effected only on the subsum including the label of the edge $\hat{e}$ of interest.

This new operator now automatically transforms a given tree-compatible factorisation of an interpolating polynomial of an idle probability tree into the tree-compatible factorisation of another (non-degenerate) probability tree which is statistically equivalent to the manipulated tree. So the two-step approach to manipulation we followed in Smith et al. (2017) where the graph $\mathcal{T}_{\hat{e}}$ needs to be transformed into a different graph $\mathcal{T}_{\not\in \hat{e}}$ can be avoided by simply evaluating this more direct algebraic manipulation.

We can now straightforwardly prove that all sensible composite manipulations in probability trees are commutative. To this end we will call two subsequent interventions *valid* if the resulting degenerate probability tree is statistically equivalent to a (non-degenerate) probability tree. We can show that such a composition is valid if and only if the two interventions are not contradictory as in Example 4.5, so if they do not force units along two mutually exclusive paths in the tree (Smith et al., 2017).

Proposition 4.7 (Composite differential manipulations)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a probability tree with graph $\mathcal{T} = (V, E)$ and interpolating polynomial $c_{\mathcal{T}}$, and let $\hat{e}, \hat{e}' \in E$. Then the intervention operator $\mathfrak{d}$ from (4.7) performs a causal intervention on an edge in the tree and is commutative, so $\mathfrak{d}(\hat{e}, c_{\mathcal{T}}) \circ \mathfrak{d}(\hat{e}', c_{\mathcal{T}}) = \mathfrak{d}(\hat{e}', c_{\mathcal{T}}) \circ \mathfrak{d}(\hat{e}, c_{\mathcal{T}})$.*

Proof. Let $\hat{e} = (\hat{v}, \hat{v}') \in E$ be an edge in the probability tree. By construction, $\mathfrak{d}(\hat{e}, c_{\mathcal{T}})$ is the direct algebraic counterpart to the intervention described in Definition 4.1 and is equivalent to the differential operator in Lemma 4.3. So $\mathfrak{d}(\hat{e}, c_{\mathcal{T}})$ performs an intervention which eliminates all unfoldings in $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ which go through the vertex $\hat{v}$ but do not pass on to $\hat{v}'$ along $\hat{e}$.

The composition $\mathfrak{d}(\hat{e}, c_{\mathcal{T}}) \circ \mathfrak{d}(\hat{e}', c_{\mathcal{T}})$ now performs two local differentiations as in (4.7). If these are not performed on the same floret then they are clearly commutative and have the same effect as the composition $\mathfrak{d}(\hat{e}', c_{\mathcal{T}}) \circ \mathfrak{d}(\hat{e}, c_{\mathcal{T}})$. □

Observe here that in case the second manipulation is enforced on a subtree which has been assigned probability zero by the first manipulation then in this algebraic framework this second manipulation has simply no effect. In the graphical framework of Definition 4.1, the second manipulation would not be well-defined because the edge it acts on would have been cancelled.

So intervention operations in staged trees rely only on a polynomial characterisation of the model in exactly the same way that the results on statistical equivalence classes of staged trees outlined in Section 3.2 are encoded using the same polynomial representation.

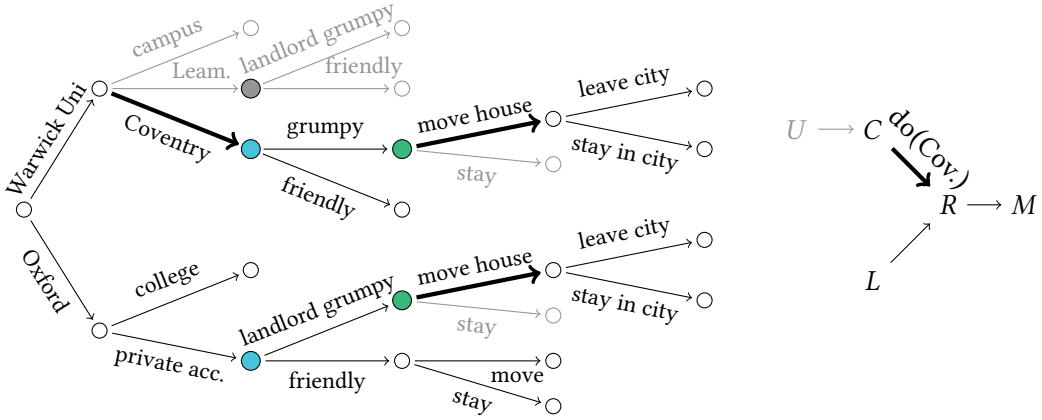
We can now use Proposition 4.7 to define composite manipulations on probability trees via differential operations. In particular, if in an X-compatible staged tree we differentiate with respect to all labels of edges $e(\hat{x_i})$ representing a state $\hat{x_i} \in \mathbf{X}_i$ then this is directly equivalent to the do-operation $\text{do}(X_i = \hat{x_i})$ from (4.1) where $X = (X_1, \ldots, X_m)$ and $i \in \{1, \ldots, m\}$.

Consider now an extension of our toy example which illustrates the advantage of defining this new operator.

Example 4.8 (Local and vertex-centred interventions)**.** Assume that the setting of Example 4.2 was part of a bigger system in which we introduce a new problem variable U representing the allocation of a student to a *university*, taking the values 'Warwick' or 'Oxford'. Students at Oxford university are assumed to live either in colleges or to rent private accommodation. In the latter case, we are again interested in the probability of a student leaving their city depending on the attitude of their landlords. The staged tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ in Fig. 4.3.1 represents this new idle system. Figure 4.3.2 shows the alternative acyclic digraph as in the example above.

In this case forcing students to live in Coventry by university policy corresponds to a local differentation of the parameter belonging to the edge labelled 'Coventry' in the staged tree, precisely as proposed in Definition 4.6. This operation again cancels out all alternative unfoldings 'campus' and 'Leamington Spa' while leaving the remaining staged tree invariant. We have hence coloured the relevant subtrees grey in the figure above whenever they disappear after projection.

Because the lower subtree in Fig. 4.3.1 was not affected by this local operation, we can now impose new policies for students from both university. Assume for instance that in the frame-



(4.3.1) A staged tree representing the model before and after intervention. **(4.3.2)** An acyclic digraph.

Figure 4.3. Two alternative graphical representations for the problem from Example 4.8, illustrating a local manipulation.

work of a mental health campaign the Student Unions of both Oxford and Warwick University lobby for students who rent with a grumpy landlord to move house. This new policy would make all students who reach a green-coloured vertex in the staged tree move along the unfolding labelled 'move'.

There are two important points to note here. First, in this context the edge intervention is especially powerful because we would not want to force students to live in Coventry if they went to Oxford university. But this is precisely what the atomic manipulation in the acyclic digraph representation of Fig. 4.3.2 would do: Pearl's atomic intervention here cancels all ancestors of the city variable and forces $C =$ 'Coventry' independent of the context. Second, thanks to the locality of the operator and the result of Proposition 4.7, we do not need to consider a temporal order when imposing these interventions and can calculate the respective effects independent of which policy has been imposed first.

4.2. Causal discovery in the CHDS

In order to assess the effect of a causal manipulation, it is necessary to know that the idle system does indeed depict a causal relationship between events. This section gives a brief introduction to methods which can enable the statistician to give a putative causal interpretation to a staged tree model.

In this context note first that the inferential methods discussed at the beginning of this chapter pertain only to situations where certain features of the distribution of the observed system—explicitly, various factorisations of marginal and conditional densities over subspaces

of the idle space, often expressed as collections of conditional independence statements over measurable variables within the space—are known, or at least plausibly conjectured. Of course this is often not the case: instead causal inference then needs to take place in two phases. The first phase, a preprocessing phase, searches over a class of models in an observed system to find within the data evidence for the types of factorisations that might be needed to make causal inferences. Only once it has been discovered that such hypotheses might be plausible can the mathematical inferences be applied to construct hypotheses of what might happen under certain treatments. In fact, this first phase of inference has been intrinsic to early causal formalisms and has in fact motivated much of its development (Spirtes et al., 1993). Model selection methods as they relate to causal inferences in standard graphical models—for example the PC algorithm developed by the authors above—have successfully been employed across a wide range of domains. However, these *causal discovery* algorithms are not currently available to search classes of causal models associated with collections of hypotheses about factorisations of probability mass functions over probability trees.

We will now make our first advances in this direction. For a given staged tree model we will draw out hypotheses which are valid for *all* members of the corresponding statistical equivalence class. In particular, using the results of Chapter 3 we can finally unambiguously state if one event always or only under certain conditions happens before another event, across all valid representations of the corresponding idle context. So in particular, we can now *discover* these invariant orders of events within a model, derived from observational data, and make causal deductions from these to generate hypotheses about the effects of controlling various parts of the system. It is central to note that this type of causal inference in staged trees has not been possible before our new development of the swap and resize operators.

The actual design of such a causal discovery algorithm for staged trees is beyond the scope of this thesis. We will thus content ourselves with presenting an analysis only of plausible causal hypotheses in the statistical equivalence class of staged trees representing the CHDS data set as presented in Section 3.2.4.

We have argued in Section 4.1 above that causes are *events* rather than random variables, and we have pointed out in Sections 1.2.3 and 3.2 that in staged tree models random variables are rather artificial notions and not as transparent as event-based semantics. Shafer (1996) makes this same point when deciding to base all causal inference methods on depictions of a system given by event trees rather than by acyclic digraphs and random variables. So the framework given by probability trees allows for a most straightforward analysis of the notion of causality.

We begin by introducing a tool which will enable us to if not determine then at least exclude certain events as possible causes of others.

Definition 4.9 (Order of events)**.** Let $\mathcal{T} = (V, E)$ be an event tree. We say that a vertex $v \in V$ is *upstream* of a vertex v' if they are connected by a root-to-leaf path, $\Lambda(v) \cap \Lambda(v') \neq \emptyset$, which passes through v before it passes through v'. Then v is also said to be *downstream* of v'.

So whenever one vertex is downstream of another vertex in a staged tree, we know that there exists a graphical representation of the underlying model which depicts the associated event $\Lambda(v)$ before $\Lambda(v')$. So this depiction can be given a chronological interpretation. However, it is necessary to know that the event $\iota_{\mathcal{T}}^{-1}(\Lambda(v)) \subseteq \Omega$ in the underlying space is *always* depicted before $\iota_{\mathcal{T}}^{-1}(\Lambda(v'))$ in all representations of the model in order to unambiguously infer that order in time from the model at hand.

Of course, temporal precedence is not sufficient for a relationship between two events to be causal. However, Pearl (2000) convincingly argues that cause and time should always follow the same directionality. So under this assumption, clearly whenever we find two inverse orders of representation of two events within the same statistical equivalence class, one is clearly *not* a cause of the other.

Now, these orderings of events in a staged tree model are most easily analysed within the polynomial equivalence class of a representation. This is because when twin structures are present, the swap operator changes the orders of edge-centred events depicted immediately up- and downstream of the twin. So by Remark 3.27, whenever two events are (conditionally) independent of each other, one is surely not a putative cause of the other. Of course an assertion like 'independent events cannot cause one another' is a rather philosophical and very debatable point. We hence use the term *putative* cause to stress that we have only found an *indication* of a causal relationship between two events, so one event can *putatively* be a cause of the other. In Pearl (2000), this terminology is given a thorough mathematical meaning

Outside of a polynomial equivalence class, the resize operator is used to contract saturated subtrees into single florets. Saturated subtrees do not contain any conditional independence information, so no assumptions which characterise the model. Any order depicted in these is thus spurious and cannot be said to have been derived from the data: compare Examples 3.32 and 3.38. So putative causal orderings between events can only be inferred in subgraphs where an order is invariant to both the swap and the resize operator. We have seen examples of this in the staged tree representing a collider graph in Example 3.35, and we will now search for similar structures in the staged tree below.

Consider again the staged tree $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ from Fig. 3.7.2 on page 106 which represents the highest scoring model for the CHDS dataset. In Section 3.2.4 and Example 3.45 from Section 3.3, we have analysed the polynomial equivalence class of this tree. All four distinct elements of this class are depicted in Figs. 3.7.2 and 3.8.1 to 3.8.3. These $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$, $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2$ and

$(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_3$ could be traversed using two swap operators $\mathfrak{s}_1$ and $\mathfrak{s}_2$. Using these swaps, we have obtained in particular the result that for wealthy children, the numbers of severe life events and hospital admissions were not ordered within the polynomial equivalence class. So following the rationale above, for this subgroup analysed in the study, life events and hospital admission cannot be said to be putative causes of each other. In terms of an interpretation of this result, we might want to argue that in wealthy families—possibly thanks to a more stable economic environment—severe life events have no impact on hospital admissions of children, and these in turn have no effect on health. In addition, we can further observe from the different orderings depicted in the staged trees $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_1$ and $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})_2$ that for children who have been admitted to hospital, wealth and life events are not ordered in time—so in the same fashion as above, these cannot be said to be putative causes of each other.

There seems to be a mechanism in the data, and evident in the stratified representations of Fig. 3.8 as analysed above, which seems to suggest that vertices associated to life events are generally downstream of vertices associated to hospital admission. In fact, there is no staged tree in the polynomial equivalence class of $(\mathcal{S}, \boldsymbol{\theta}_{\mathcal{S}})$ that would allow for the total order of life events happening before hospital admission. This is because no composition of the swaps $\mathfrak{s}_1$ and $\mathfrak{s}_2$ can form a level-swap on these staged trees. So a model which treats life events as an explanatory variable of the response variable hospital admission as in the study Barclay et al. (2013) is less supported by the data than one treating hospital admissions as an explanatory variable of life events as in Cowell and Smith (2014). See also Smith et al. (2017) for a comparison of these two studies.

Of course this deduction needs the caveat that there exists a reasonably high scoring staged tree model which *does* embed this reversal (Cowell and Smith, 2014). So evidence for the chosen order is quite weak. However, it is nevertheless *formally suggested* in the unambiguous way we discuss above. Note that no deductions about an ordering of variables were possible within the original Bayesian network model for these data because the highest scoring model turns out to be decomposable. This demonstrates that the extra structure of the staged tree enables us to draw out new potential causal hypotheses that could not be discovered when using more conventional graphical methods.

Conclusions

In this thesis we have developed a new formalism around staged tree models which draws on techniques from graph theory, computer science and algebraic geometry. We have seen that this formalism enables us to answer a number of formerly open queries in these parametric and discrete statistical models. Graph theory, especially the properties of event trees and the way in which these depict a pre-ordered set of vertices, enabled us to make very precise statements about the interplay of events within a discrete unfolding. This is because special nested structures within such a graph relate to a pre-order on the sigma-algebra of events underlying the model represented by that graph. Labelling event trees, we then opened the door to employ techniques from computer science and symbolic computing. In fact, in defining an interpolating polynomial in the edge labels of an event tree we could both link our graphical model representations to well-known arithmetic circuits and employ these in probabilistic queries, using a differential framework. That same polynomial could then be used as a surrogate to a labelled event tree. Its properties—in terms of a representation in a nested, bracketed form, and hence in terms of the interplay of common divisors between terms in the polynomial—turned out to be sufficiently strong to enable us to determine the class of all labelled event trees which code the same assumptions in an inferential context. This class could be traversed using intuitive graphical operations, justified by their analogues in terms of summation, multiplication and substitution operations. The algebraic framework of which this polynomial is formally an element further enabled us to embed our work into the very successful research area of Algebraic Statistics and provided the tools for both specifying a staged tree model as a solution set of polynomial equations—interestingly given by odds ratios—and for drawing them as images of a parametrisation map inside a probability simplex. Both the algebraic and the symbolic approach to staged tree models have further enabled us to develop a new set of tools to be used in a causal interpretation of these models and of employing new techniques for measuring the effect of causal manipulations or controls.

Now that this new interdisciplinary—at least within mathematics highly interdisciplinary—framework has been successfully established, we have naturally collected a number of new queries on the way which will hopefully be answered during future research. We present these below in the order in which they appear in the chapters above.

A first stream of research comes directly out of our result that staged tree models can be characterised as solution sets of a collection of odds ratios together with positivity constraints and sum-to-1 conditions. Whilst this result had both an interesting statistical interpretation and enabled us to draw a number of insightful illustrations of small models, it can only be a very first step into a rich geometric analysis of the properties of these models. In particular, we note that the analogous characterisation of Bayesian networks allowed for an interesting result also in algebro-geometric terms: decomposable models correspond to toric varieties. These objects have a number of interesting geometric properties. So are there any equally interesting geometric properties of the varieties we determined for staged tree models? Can characteristics of the semi-algebraic sets we found be linked back to statistical properties of the model? An analysis of this type would further enable us to understand the interplay between sum-to-1 conditions and a specification of our models and of the effect of imposing positivity conditions. The latter would be especially interesting in an analysis of the behaviour of our models on the boundary where probabilities can be zero or one: that this is a subtle point in statistical inference is well known. In addition, a precise understanding of the effect of ignoring these conditions would foster a better understanding of the loss of information we face when applying a brute force approach to implicitisation as we have done above. So we might want to know what the difference is between a model specified as the image of a parametrisation map and the varieties we obtain when allowing for the domain and co-domain of that map to be a high-dimensional Euclidian space rather than a probability simplex. All of these directions of future research require advanced algebraic techniques, as we have outlined above, and have not been the scope of this thesis. However, because the research area of Algebraic Statistics has over the past decade contributed a number of elegant links between formerly disconnected areas of mathematics and has provided fruitful results in both algebraic and statistical terms, we are confident that with the right tools a proper understanding of the algebraic geometry underlying staged tree models has the potential to make a huge contribution to the current literature.

A second open question concerns computational complexity of the results we presented in this thesis. We have here been able to give rough answers to the most pressing questions: namely those concerning an application of the swap operator and those concerning the use of the differential framework. The computations underlying a traversing of a statistical equivalence class of staged trees via their interpolating polynomials are based on computations of primary decompositions of monomials ideals. These are again well studied and implemented in a wide range of freely available software. In addition, the use of a network polynomial in a differential framework is particularly simple because every probabilistic query we analysed here can be answered by a compilation of that polynomial into an arithmetic circuit, or more

generally into what is known as a *sum-product net*. These are well-understood tools in computer science with readily available and fast algorithms. However, we have here contented ourselves with a discovery of these links to established computational mechanisms without providing an analysis of how these tools might be adapted to the generality of our models over their usual applications. A future implementation of these methods as tailored to staged tree models will need to provide bounds on this computational complexity in terms of the size of a given staged tree[27]. Then in order for our algorithms to be implementable at least a naïve count of operations will need to be provided. In addition, once we have developed an algorithm for traversing the statistical equivalence class of a given staged tree, we will need to provide an interface[28] between our computations in computer algebra software to statistical software such as `R` (R Core Team, 2016). In this way, whenever data are available, we will be able to make use of our results to significantly speed up model selection techniques: for instance, by applying scoring rules directly to equivalence classes of staged trees—picking one representative of a potentially huge statistical equivalence class—rather than to single staged trees many of which will represent the same model.

The third aspect of staged tree models we have been able to shed some new light on in this thesis is a causal interpretation of a given graphical representation. Whilst the computation of effects of causal controls is reasonably well understood—both in algebraic terms and in a differential framework—it remains difficult to assert when exactly it is unambiguous to talk about a model being causal or a graphical model representation depicting a causal relationship. We have been able to make first advances in this direction by determining conditions under which the relationships between certain events cannot be said to be causal. However this is clearly not sufficient. So the development of causal discovery algorithms for staged tree models is an important and, at the time of writing, open field of research. We believe that in such a development it would be fruitful to first determine new subclasses of classes of statistically equivalent staged trees, namely those which can in a yet to be formalised sense be said to be 'causally equivalent'. These causal equivalence classes can then be analysed in terms of their interpolating polynomials and algebraic interpretation as above. It would not be surprising to find that classes of staged trees which do not allow for certain reorderings in their depiction of events—and hence reorderings of the terms and labels within their interpolating polynomial—also share a common geometric interpretation. So the design of a new causal discovery algorithm over staged trees and chain event graphs would draw again on the computer science or symbolic approach above, on graph theory and on algebraic geometry: all areas we have

27 It is for instance known that the number of possible staged trees grows very fast in the number of atoms: in fact, this growth is intrinsically linked to the *Bell number* (Smith et al., 2017).

28 We note that there are `R`-packages such as `algstat` available which might be employed in this development.

drawn together in this thesis. As a consequence, this work provides a most promising basis from which we can now develop new and powerful causal inference techniques for staged tree models.

A. Proofs

In this appendix we provide the proofs of three results which have been important to the development presented in this thesis but have—for reasons of text flow—not found their way into the main body of the text.

Proposition A.1 (See pages 37 and 64)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a saturated (uncoloured staged) tree with $n = \#\Lambda(\mathcal{T})$ root-to-leaf paths. Then the associated model $\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} = \Delta^{\circ}_{n-1}$ is equal to the open $n-1$-dimensional probability simplex.*

Proof. We show the claim proving two set inclusions.

'$\mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})} \subseteq \Delta^{\circ}_{n-1}$' Every element in the staged tree model is a vector with n components. By Proposition 1.6, the components of this vector are positive and sum to unity. They are thus elements of the probability simplex of dimension $n-1$.

'$\Delta^{\circ}_{n-1} \subseteq \mathbb{P}_{(\mathcal{T},\boldsymbol{\theta}_{\mathcal{T}})}$' Denote by $\Psi_{\mathcal{T}} : \bigtimes_{j=1}^{m} \Delta^{\circ}_{n_j-1} \to \Delta^{\circ}_{n-1}$ the tree parametrisation of a saturated tree $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ with graph $\mathcal{T} = (V, E)$. Then the claim above is equivalent to the proposition that this parametrisation is surjective onto the simplex, so $\Psi_{\mathcal{T}}(\bigtimes_{j=1}^{m} \Delta^{\circ}_{n_j-1}) = \Delta^{\circ}_{n-1}$. We show below that this is true.

Let thus $\boldsymbol{p} = (p_1, \ldots, p_n) \in \Delta^{\circ}_{n-1}$ be any point inside the probability simplex. We will prove that there exists a vector $\boldsymbol{\theta} \in \times_{j=1}^{m} \Delta^{\circ}_{n_j-1}$ in the preimage of the tree parametrisation which is mapped to that point, so $\Psi_{\mathcal{T}}(\boldsymbol{\theta}) = \boldsymbol{p}$. Denote therefore for every vertex $v \in V$ in the tree by $I(v) \subseteq \{1, \ldots, n\}$ the index sets belonging to root-to-leaf paths in the associated vertex-centred event, such that $\Lambda(v) = \{\lambda_i \mid i \in I(v)\}$. For every edge $(v, v') \in E$, we now set the label of that edge to be

$$\theta(v, v') = \frac{\sum_{j \in I(v')} p_j}{\sum_{i \in I(v)} p_i}. \tag{A.1}$$

Then every such label is a number strictly between zero and one, the sum of all the labels which belong to the same vector of floret labels is equal to one, and the products of these labels along root-to-leaf paths are equal to components of the point $\boldsymbol{p}$. We now show that this is so.

In fact, observe first that because for an edge (v, v') one index set $I(v')$ is always contained in the other index set, $I(v') \subseteq I(v)$, the fraction above is smaller than 1, and that because $\boldsymbol{p}$ has only positive components which sum to unity, the fraction (A.1) is indeed a number

$\theta(v, v') \in (0, 1)$ strictly between zero and one. In addition, these labels fulfil the sum-to-1 condition across florets

$$\sum_{v' \in \mathrm{ch}(v)} \theta(v, v') = \frac{1}{\sum_{i \in I(v)} p_i} \sum_{v' \in \mathrm{ch}(v)} \sum_{j \in I(v')} p_j = 1 \tag{A.2}$$

because every event centred at a vertex can be written as the disjoint union of events centred at that vertex' children, so $\bigcup_{v' \in \mathrm{ch}(v)} I(v') = I(v)$. Thus, choosing the labels $\boldsymbol{\theta} = \big(\theta(e) \mid e \in E\big)$ of the saturated tree as in (A.1) we obtain indeed a vector in the product of probability simplices $\boldsymbol{\theta} \in \bigtimes_{j=1}^{m} \Delta^{\circ}_{n_j - 1}$ which is the domain of the tree parametrisation: so this is an amenable choice of labels. In a second step, we now need to make sure this choice of labels is mapped to the correct point.

In fact, along every root-to-leaf path $\lambda_i = ((v_{i1}, v_{i2}), (v_{i2}, v_{i3}) \ldots, (v_{ik_i - 1}, v_{ik_i})) \in \Lambda(\mathcal{T})$ in the tree the product of edge labels now simplifies to

$$\begin{aligned} \prod_{j=1}^{k-1} \theta(v_{ij}, v_{ij+1}) &= \frac{\sum_{s_{i2} \in I(v_{i2})} p_{s_{i2}}}{\sum_{r_{i1} \in I(v_{i1})} p_{r_{i1}}} \cdot \frac{\sum_{s_{i3} \in I(v_{i3})} p_{s_{i3}}}{\sum_{r_{i2} \in I(v_{i2})} p_{r_{i2}}} \cdots \frac{\sum_{s_{ik_i} \in I(v_{ik_i})} p_{s_{ik_i}}}{\sum_{r_{ik_i - 1} \in I(v_{ik_i - 1})} p_{r_{ik_i - 1}}} \\ &= \frac{p_i}{\sum_{r=1}^{n} p_r} = p_i \end{aligned} \tag{A.3}$$

because all root-to-leaf paths go through the root, so $I(v_{i1}) = \{1, \ldots, n\}$, and only the i^{th} path ends in v_{ik_i}, so $I(v_{ik_i}) = \{i\}$, for all $i = 1, \ldots, n$. So by construction, the image $\Psi_{\mathcal{T}}(\boldsymbol{\theta}) = \boldsymbol{p}$ of these labels is indeed the generic point we chose inside the probability simplex. The claim follows. □

Proposition A.2 (See page 83)**.** *Let $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ be a square-free staged tree and let $\pi_{\boldsymbol{\theta}, \mathcal{T}}$ denote an atomic monomial as in (1.12). Then the map $\pi_{\boldsymbol{\theta}, \mathcal{T}} : \lambda \mapsto \prod_{\lambda \in \Lambda(\mathcal{T})} \theta(e)$ is injective.*

Proof. Assume by contradiction that $\pi_{\boldsymbol{\theta}, \mathcal{T}}$ was not injective. Then there exist two root-to-leaf paths $\lambda, \lambda' \in \Lambda(\mathcal{T})$ which are not equal and for which the monomials $\pi_{\boldsymbol{\theta}, \mathcal{T}}(\lambda) = \pi_{\boldsymbol{\theta}, \mathcal{T}}(\lambda')$ are equal independent of an assignment of values to the labels $\boldsymbol{\theta}$.

Denote these two root-to-leaf paths by $\lambda = (e_1, \ldots, e_k)$ and $\lambda = (e'_1, \ldots, e'_l)$ and denote by v the vertex at which they split: so the initial sequence of s edges $(e_1, \ldots, e_{s-1}) = (e'_1, \ldots, e'_{s-1})$ is the same for both subpaths from v_0 up to v and then differs in the floret $\mathcal{F}_v$ such that $e_s \neq e'_s$. Of course, $s = 0$ and$v = v_0$ is possible. Because now the root-to-leaf paths pass along two edges which are elements of the same floret, $e_s, e'_s \in E(v)$, but are different, these edges also have different labels: $\theta(e_s) \neq \theta(e'_s)$.

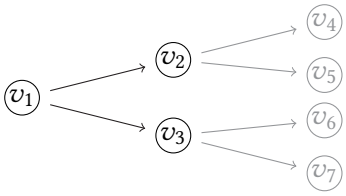


Figure A.1. A binary event tree as in Proposition A.3. The black coloured graph has one level $l = 1$ and $n_l = 2$ root-to-leaf paths with a total of $\#V_l = 3$ vertices. When adding an extra grey-coloured level, we obtain $l + 1 = 2$ levels and twice the number $n_{l+1} = 4 = 2n_l$ of root-to-leaf paths with $\#V_{l+1} = 7$ vertices. Note that $\#V_{l+1} = 2n_{l+1} - 1$ as claimed.

In order for the two corresponding atomic monomials to coincide $\prod_{i=1}^{k} \theta(e_i) = \prod_{j=1}^{l} \theta(e'_j)$, the root-to-leaf path λ has to subsequently pass through an edge labelled $\theta(e'_s)$ and, in analogy, λ' has to pass through an edge labelled $\theta(e_s)$ within $\mathcal{T}(v)$. Hence, because $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is a staged tree, the floret labels $\boldsymbol{\theta}_v$ need to be repeated within the induced subtree $(\mathcal{T}(v), \boldsymbol{\theta}_{\mathcal{T}(v)})$ emanating from v. As a consequence, there is a vertex in $\mathcal{T}(v)$ which is not equal to v but is connected to v by a path *and* is in the same stage as v: so by Definition 1.11, $(\mathcal{T}(v), \boldsymbol{\theta}_{\mathcal{T}(v)})$ is not square-free. This implies that $(\mathcal{T}, \boldsymbol{\theta}_{\mathcal{T}})$ is not square-free—a contradiction. □

Proposition A.3 (See page 111.)**.** *Let $\mathcal{T} = (V, E)$ be a binary event tree, that is every non-leaf vertex has precisely two emanating edges: so $\#E(v) = 2$ for these $v \in V$. Assume in addition that all root-to-leaf paths in the event tree have the same fixed number of edges, so $E(\lambda) = l$ for all $\lambda \in \Lambda(\mathcal{T})$ and some $l \in \mathbb{N}$.*

Denote by $n \in \mathbb{N}$ the number of root-to-leaf paths of the tree, so $\#\Lambda(\mathcal{T}) = n$. Then the number of vertices in the tree equals two times the number of root-to-leaf paths less one: $\#V = 2n - 1$.

Proof. We prove this result by induction over the number of levels $l \in \mathbb{N}$ of an event tree. Denote hence by V_l the vertex set of a binary event tree as above with l levels and n_l root-to-leaf paths, $l \in \mathbb{N}$.

Let $l = 1$: Then the event tree is a single floret with one root and two children which are leaves, and two edges. So $\#V_1 = 3$, $n_1 = 2$ and $\#V_1 = 3 = 2 \cdot 2 - 1 = 2n_1 - 1$. Consider Fig. A.1 for an illustration. So the claim is true in this case.

Let now $l \in \mathbb{N}$ be fixed but arbitrary. We assume that the proposition is true for l and show validity of the result for $l + 1$. Note first that when adding an extra level to a binary tree as above, we add one binary floret to every leaf. This doubles the number of root-to-leaf paths,

so $n_{l+1} = 2n_l$. Then the induction hypothesis that $\#V_l = 2n_l - 1$ implies that

$$\#V_{l+1} = \#V_l + 2n_l = 2n_l - 1 + 2n_l = 2(2n_l) - 1 = 2n_{l+1} - 1. \tag{A.4}$$

The claim follows. □

Bibliography

Abbott, J., A. M. Bigatti, and G. Lagorio (2016). CoCoA-5: a system for doing Computations in Commutative Algebra. Available at `http://cocoa.dima.unige.it`. (Cited on pages x and 36.)

Altham, P. M. E. (1969). Exact Bayesian analysis of a 2×2 contingency table, and Fisher's "exact" significance test. *J. Roy. Statist. Soc. Ser. B 31*, 261–269. (Cited on page 48.)

Altham, P. M. E. (1970a). The measurement of association in a contingency table: three extensions of the cross-ratios and metrics methods. *J. Roy. Statist. Soc. Ser. B 31*, 395–407. (Cited on page 48.)

Altham, P. M. E. (1970b). The measurement of association of rows and columns for an $r \times s$ contingency table. *J. Roy. Statist. Soc. Ser. B 32*, 63–73. (Cited on page 48.)

Andersson, S. A., D. Madigan, and M. D. Perlman (1997). A characterization of Markov Equivalence Classes for Acyclic Digraphs. *Ann. Statist. 25*(2), 505–541. (Cited on page 81.)

Barclay, L. M., J. L. Hutton, and J. Q. Smith (2013). Refining a Bayesian network using a Chain Event Graph. *Internat. J. Approx. Reason. 54*(9), 1300–1309. (Cited on pages iii, 25, 29, 30, 105, and 131.)

Benedetti, R. and J.-J. Risler (1990). *Real algebraic and semi-algebraic sets*. Actualités Mathématiques. [Current Mathematical Topics]. Hermann, Paris. (Cited on page 55.)

Berzuini, C., P. Dawid, and L. Bernardinell (2012). *Causality: Statistical Perspectives and Applications*. Wiley Series in Probability and Statistics. Wiley. (Cited on page 120.)

Blitzstein, J. and J. Hwang (2014). *Introduction to Probability*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press. (Cited on pages 1, 9, and 72.)

Boutilier, C., N. Friedman, M. Goldszmidt, and D. Koller (1996). Context-specific independence in Bayesian networks. In *Uncertainty in artificial intelligence (Portland, OR, 1996)*, pp. 115–123. Morgan Kaufmann, San Francisco, CA. (Cited on page 25.)

Casella, G. and R. L. Berger (2002). *Statistical Inference* (2nd ed.). Duxbury advanced series in statistics and decision sciences. Thomson Learning. (Cited on page 84.)

Castillo, E., J. M. Gutiérrez, and A. S. Hadi (1995). Parametric structure of probabilities in Bayesian networks. In *Symbolic and quantitative approaches to reasoning and uncertainty (Fribourg, 1995)*, Volume 946 of *Lecture Notes in Comput. Sci.*, pp. 89–98. Springer, Berlin. (Cited on page 73.)

Chan, H. and A. Darwiche (2002). When do numbers really matter? *J. Artificial Intelligence Res. 17*, 265–287 (electronic). (Cited on page 73.)

CHDS (2017). Christchurch Health and Development Study. Online resource available at `http://www.otago.ac.nz/christchurch/research/healthdevelopment`. Accessed February 8, 2017. (Cited on page 105.)

Collazo, R. A. and J. Q. Smith (2015). A New Family of Non-Local Priors for Chain Event Graph Model Selection. *Bayesian Anal. 11*(4), 1165–1201. (Cited on pages 29 and 32.)

Constantinou, P. and P. A. Dawid (2015). Extended conditional independence and applications in causal inference. Available at `arXiv:1512.00245v1`. (Cited on page 120.)

Coudouel, A., J. S. Hentschel, and Q. T. Wodon (2002). *Poverty Measurement and Analysis*, pp. 27–74. The World Bank. (Cited on page 106.)

Cowell, R. G. and J. Q. Smith (2014). Causal discovery through MAP selection of stratified Chain Event Graphs. *Electron. J. Stat. 8*(1), 965–997. (Cited on pages 29, 33, 105, 106, and 131.)

Cox, D. A., J. Little, and D. O'Shea (2015). *Ideals, varieties, and algorithms* (Fourth ed.). Undergraduate Texts in Mathematics. Springer, Cham. An introduction to computational algebraic geometry and commutative algebra. (Cited on pages 34, 35, 36, 54, 55, 57, 58, and 113.)

Darwiche, A. (2003). A differential approach to inference in Bayesian networks. *J. ACM 50*(3), 280–305 (electronic). (Cited on pages 72, 73, 74, 79, and 120.)

Darwiche, A. (2009). *Modeling and reasoning with Bayesian networks*. Cambridge University Press, Cambridge. (Cited on pages 72, 78, and 79.)

Drton, M., B. Sturmfels, and S. Sullivant (2009). *Lectures on algebraic statistics*, Volume 39 of *Oberwolfach Seminars*. Birkhäuser Verlag, Basel. (Cited on pages 10, 36, 48, and 62.)

Drton, M. and S. Sullivant (2007). Algebraic Statistical Models. *Statist. Sinica 17*(4), 1273–1297. (Cited on pages 53, 55, 56, and 58.)

Fergusson, D. M., L. J. Horwood, and F. T. Shannon (1986). Social and family factors in childhood hospital admission. *Journal of Epidemiology and Community Health 40*, 50–58. (Cited on pages 29 and 105.)

Freeman, G. and J. Q. Smith (2011). Bayesian MAP model selection of Chain Event Graphs. *J. Multivariate Anal. 102*(7), 1152–1165. (Cited on page 32.)

Garcia, L. D., M. Stillman, and B. Sturmfels (2005). Algebraic geometry of Bayesian networks. *J. Symbolic Comput. 39*(3-4), 331–355. (Cited on pages 36 and 54.)

Garthwaite, P. H., J. B. Kadane, and A. O'Hagan (2005). Statistical method for eliciting probability distributions. *J. Amer. Statist. Assoc. 100*(470), 680–700. (Cited on page 48.)

Geiger, D., C. Meek, and B. Sturmfels (2006). On the toric algebra of graphical models. *Ann. Statist. 34*(3), 1463–1492. (Cited on pages 13, 36, 39, and 54.)

Gibilisco, P., E. Riccomagno, M. P. Rogantin, and H. P. Wynn (Eds.) (2010). *Algebraic and geometric methods in statistics*. Cambridge University Press, Cambridge. (Cited on page 36.)

Görgen, C., M. Leonelli, and J. Q. Smith (2015). A Differential Approach for Staged Trees. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Volume 9161 of *Lecture Notes in Computer Science*, pp. 346–355. (Cited on pages ix, 6, 21, 26, 73, 74, and 79.)

Görgen, C. and J. Q. Smith (2015). Equivalence Classes of Staged Trees. Available at `arXiv:1512.00209v2[math.ST]`. (Cited on pages ix, 6, 14, 17, and 80.)

Görgen, C. and J. Q. Smith (2016). A differential approach to causality in staged trees. In *Proceedings of the Eighth International Conference on Probabilistic Graphical Models*, Volume 52 of *JMLR Workshop and Conference Proceedings*, pp. 207–215. (Cited on pages ix, 6, 120, and 126.)

Görgen, C., J. Q. Smith, E. Riccomagno, and A. Bigatti (2017). Discovering statistical equivalence classes of discrete statistical models using computer algebra. In preparation. (Cited on pages ix, 6, 54, 86, 88, 109, 113, and 115.)

Heckerman, D. (1998). *A Tutorial on Learning with Bayesian Networks*, pp. 301–354. The MIT Press. (Cited on page 81.)

Hoşten, S. and S. Sullivant (2002). Gröbner bases and polyhedral geometry of reducible and cyclic models. *J. Combin. Theory Ser. A 100*(2), 277–301. (Cited on page 62.)

Jaeger, M. (2004). Probabilistic decision graphs—combining verification and AI techniques for probabilistic inference. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 12*(January 2004, suppl.), 19–42. New trends in probabilistic graphical models. (Cited on page 19.)

Jensen, F. V. and F. Jensen (1994). Optimal Junction Trees. In M. Kaufmann (Ed.), *Proceedings of the 10th Conference on Uncertainty in Artifical Intelligence*, Volume 10, San Mateo, CA. (Cited on page 22.)

Lauritzen, S. L. (1996). *Graphical models*, Volume 17 of *Oxford Statistical Science Series*. The Clarendon Press, Oxford University Press, New York. Oxford Science Publications. (Cited on pages 5, 9, 17, 19, 21, 23, and 81.)

Lauritzen, S. L. and T. S. Richardson (2002). Chain graph models and their causal interpretations. *J. R. Stat. Soc. Ser. B Stat. Methodol. 64*(3), 321–361. (Cited on page 22.)

Lauritzen, S. L. and D. J. Spiegelhalter (1988). Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *J. Roy. Statist. Soc. Ser. B 50*(2), 157–224. With discussion. (Cited on page 86.)

Leonelli, M., C. Görgen, and J. Q. Smith (2015). Sensitivity analysis, multilinearity and beyond. Available at `arXiv:1512.02266[cs.AI]`. (Cited on pages ix, 12, and 73.)

Maathuis, M. H. and D. Colombo (2015). A generalized back-door criterion. *Ann. Statist. 43*(3), 1060–1088. (Cited on page 120.)

Maruri-Aguilar, H., E. Sáenz-de Cabezón, and H. P. Wynn (2013). Alexander duality in experimental designs. *Ann. Inst. Statist. Math. 65*(4), 667–686. (Cited on page 36.)

McAllester, D., M. Collins, and F. Pereira (2008). Case-factor diagrams for structured probabilistic modeling. *J. Comput. System Sci. 74*(1), 84–96. (Cited on page 19.)

Miller, E. and B. Sturmfels (2005). *Combinatorial Commutative Algebra*. Graduate Texts in Mathematics. Springer New York. (Cited on page 113.)

Mond, D., J. Smith, and D. van Straten (2003). Stochastic factorizations, sandwiched simplices and the topology of the space of explanations. *R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci. 459*(2039), 2821–2845. (Cited on page 36.)

Pachter, L. and B. Sturmfels (2005). *Algebraic statistics for computational biology*. Cambridge University Press, New York. (Cited on pages 36 and 58.)

Pearl, J. (1993). Comment: Graphical Models, Causality and Intervention. *Statist. Sci.* (8), 266–269. (Cited on page 119.)

Pearl, J. (1995a). Causal diagrams for empirical research. *Biometrika 82*(4), 669–710. With discussion and a rejoinder by the author. (Cited on page 119.)

Pearl, J. (1995b). Causal diagrams for empirical research. *Biometrika 82*(4), 669–710. With discussion and a rejoinder by the author. (Cited on page 121.)

Pearl, J. (2000). *Causality* (First ed.). Cambridge University Press, Cambridge. Models, reasoning, and inference. (Cited on pages 7, 119, 120, 121, 124, and 130.)

Pearl, J. (2009). Causal inference in Statistics: An overview. *Stat. Surv. 3*, 96–146. (Cited on page 119.)

Pistone, G., E. Riccomagno, and H. P. Wynn (2001a). *Algebraic Statistics*, Volume 89 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, Boca Raton, FL. Computational commutative algebra in statistics. (Cited on pages 12, 36, and 83.)

Pistone, G., E. Riccomagno, and H. P. Wynn (2001b). Gröbner bases and factorisation in discrete probability and Bayes. *Stat. Comput. 11*(1), 37–46. (Cited on page 72.)

Pistone, G., E. Riccomagno, and H. P. Wynn (2006). A note on computation algebra for discrete statistical models. In B. Hanzon and M. Hazewinkel (Eds.), *Proceedings of the KNAW academic colloquium "Constructive Algebra and System Theory"*, pp. 341–347. Royal Netherlands Academy of Arts and Sciences. (Cited on pages 36 and 54.)

R Core Team (2016). R: A language and environment for statistical computing. Available at `https://www.R-project.org`. (Cited on page 135.)

Salmerón, A., A. Cano, and S. Moral (2000). Importance sampling in Bayesian networks using probability trees. *Comput. Statist. Data Anal. 34*(4), 387–413. (Cited on page 19.)

Schachter, R. D. (1988). Probabilistic Inference and Influence Diagrams. *Operations Research 36*(4), 589–605. (Cited on page 93.)

Settimi, R. and J. Q. Smith (2000). Geometry, moments and conditional independence trees with hidden variables. *Ann. Statist. 28*(4), 1179–1205. (Cited on page 36.)

Shachter, R. (1998). Probabilistic inference and influence diagrams. *Operations Research 36*(4). (Cited on page 19.)

Shafer, G. (1996). *The Art of causal Conjecture*. MIT Press, Cambridge. (Cited on pages 14, 15, 17, 120, and 129.)

Silva, R. and R. Evans (2014). Causal inference through a witness protection program. Available at `arXiv:1406.0531v2`. (Cited on page 120.)

Smith, J. Q. (2010). *Bayesian Decision Analysis, Principles and Practice*. Cambridge University Press. (Cited on pages 9, 25, 26, 33, 48, and 100.)

Smith, J. Q. and P. E. Anderson (2008). Conditional independence and Chain Event Graphs. *Artificial Intelligence 172*(1), 42–68. (Cited on pages 1, 4, 17, 19, 21, 24, 26, 30, and 32.)

Smith, J. Q., C. Görgen, and R. A. Collazo (2017). *Chain Event Graphs*. In preparation for Chapman & Hall/CRC, Boca Raton, FL. (Cited on pages x, 16, 20, 21, 22, 26, 28, 33, 80, 93, 97, 105, 121, 126, 131, and 135.)

Spirtes, P., C. Glymour, and R. Scheines (1993). *Causation, Prediction, and Search* (1st ed.). MIT press. (Cited on pages 119 and 129.)

Studený, M. (2005). *Probabilistic Conditional Independence Structures*. Information Science and Statistics. Springer, London. (Cited on page 21.)

Sturmfels, B. (1996). *Gröbner bases and convex polytopes*, Volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI. (Cited on pages 36 and 58.)

PGF Ti*k*Z (2010). *TikZ & PGF (version 2.10)*. Developed by Till Tantau et al. Online resource available at `http://sourceforge.net/projects/pgf`. Accessed Feburary 22, 2017. (Cited on page x.)

Thwaites, P. (2013). Causal Identifiability via Chain Event Graphs. *Artificial Intelligence 195*, 291–315. (Cited on pages 33, 120, and 121.)

Thwaites, P. A. and J. Q. Smith (2015a). A New Method for tackling Asymmetric Decision Problems. In *Proceedings of the 10th Workshop on Uncertainty Processing (WUPES'15)*, pp. 179–190. (Cited on page 33.)

Thwaites, P. A. and J. Q. Smith (2015b). A Separation Theorem for Chain Event Graphs. Available at `arXiv:1501.05215`. (Cited on pages 6, 16, 25, 26, 33, 81, 93, and 97.)

Thwaites, P. A., J. Q. Smith, and R. G. Cowell (2008). Propagation using Chain Event Graphs. In *Proceedings of the 24th Conference on Uncertainty in Artifical Intelligence*, Helsinki, pp. 546–553. (Cited on page 33.)

Thwaites, P. A., J. Q. Smith, and E. Riccomagno (2010). Causal analysis with Chain Event Graphs. *Artificial Intelligence 174*(12-13), 889–909. (Cited on pages 33 and 120.)

Wolfram Research Inc. (2016). *Mathematica 11.0*. (Cited on pages x and 68.)

Zwiernik, P. (2016). *Semialgebraic statistics and latent tree models*, Volume 146 of *Monographs on Statistics and Applied Probability*. Chapman & Hall/CRC, Boca Raton, FL. (Cited on pages 19 and 36.)